AddTag, a two-step approach with supporting software package that facilitates CRISPR/Cas-mediated precision genome editing

Supplementary material

Thaddeus D. Seher, Namkha Nguyen, Diana Ramos, Priyanka Bapat, Clarissa J. Nobile, Suzanne S. Sindi, and Aaron D. Hernday

# Supplemental tables

## Table S1 – Comparison of gRNA design software

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name (CITATION) | Spacer motifs | 3’-adjacent PAM motifs | 5’-adjacent PAM motifs | Command line | Web | Interactive  output | Integrated  ranking | Scoring algorithms | Genomes | cPCR Primers | Ambiguous bases | Knock-in design |
| [AddTag](https://github.com/tdseher/addtag-project) (This study) | Any | Any | Any | ✓ | ✗ | ✗ | ✓ | 13 | Any | ✓ | ✓ | ✓ |
| [Benchling](https://benchling.com/) (2019) | N{16,24} | N{2,8} | N{2,8} | ✗ | ✓ | ✓ | ✗ | 2 | 163 | Limited | ✗ | ✗ |
| [Breaking-Cas](https://bioinfogp.cnb.csic.es/tools/breakingcas/) (Oliveros *et al.* 2016) | N{18,25} | Any | Any | ✗ | ✓ | ✓ | ✗ | 1 | 1457 | ✗ | Limited | ✗ |
| [Cas9 Target Finder](https://shigen.nig.ac.jp/fly/nigfly/cas9/cas9TargetFinder.jsp) (Kondo and Ueda 2013) | GN{19} | NGG | ✗ | ✗ | ✓ | ✗ | ✗ | 1 | ✗ | ✗ | Limited | ✗ |
| [Cas-Designer](http://www.rgenome.net/cas-designer/) (Park *et al.* 2015) | Any | Any | Any | ✓ | ✓ | ✗ | ✗ | 4 | Any\* | ✗ | Limited | ✗ |
| [CasFinder](http://arep.med.harvard.edu/CasFinder/) (Aach *et al.* 2014) | N{1,} | Any | ✗ | ✓ | ✗ | ✗ | ✗ | 1 | Any | ✗ | Limited | ✗ |
| [CCTOP](https://cctop.cos.uni-heidelberg.de/) (Stemmer *et al.* 2015; Stemmer *et al.* 2017) | Any | Any | ✗ | ✓ | ✓ | ✗ | ✗ | 2 | Any | ✗ | Limited | ✗ |
| [CHOPCHOP](http://chopchop.cbu.uib.no/) (Montague *et al.* 2014; Labun *et al.* 2016) | N{1,}\* | Any\* | Any\* | ✓ | ✓ | ✓ | ✓ | 9 | Any\* | ✓ | Limited | ✗ |
| [CRISPOR](http://crispor.tefor.net/) (Haeussler *et al.* 2016) | N{20,24) | 20 | 13 | ✗ | ✓ | ✓ | ✗ | 16 | Any\* | ✓ | ✗ | ✗ |
| [CRISPR4P](https://github.com/Bahler-Lab/crispr4p) (Rodríguez-López *et al.* 2017) | N{20} | NGG | ✗ | ✓ | ✓ | ✗ | ✗ | 0 | 1 | ✓ | ✗ | ✓ |
| CRISPR Design Tool & CRISPR Specificity Analysis | N{18,22} | Any | Any | ✗ | ✓ | ✓ | ✗ | 1 | 39 | ✗ | ✗ | ✗ |
| [CRISPRdirect](http://crispr.dbcls.jp/) (Naito *et al.* 2014) | N{20} | Any | ✗ | ✗ | ✓ | ✓ | ✓ | 4 | 671 | ✗ | ✗ | ✗ |
| [CRISPR-P](http://crispr.hzau.edu.cn/CRISPR2/) (Lei *et al.* 2014; Liu *et al.* 2017) | N{20,22} | 6 | 8 | ✗ | ✓ | ✓ | ✗ | 4 | 75 plants | ✗ | ✗ | ✗ |
| [CRISPRscan](https://www.crisprscan.org/) (Moreno-Mateos *et al.* 2015) | 2 | NGG | 2 | ✗ | ✓ | ✗ | ✗ | 2 | 14 animals | ✗ | ✗ | ✗ |
| [GuideScan](http://www.guidescan.com/) (Perez *et al.* 2017) | N{20} | NGG | TTTN | ✓ | ✓ | ✗ | ✗ | 2 | Any\* | ✗ | Limited | ✗ |
| [sgRNA Design Tool](https://portals.broadinstitute.org/gpp/public/analysis-tools/sgrna-design) (Doench *et al.* 2014; Broad Institute 2015) | 2 | 2 | 2 | ✗ | ✓ | ✗ | ✓ | 6 | 4 | ✗ | ✗ | ✗ |
| [Synthego CRISPR Design Tool](https://design.synthego.com/#/) | N{20} | NGG | ✗ | ✗ | ✓ | ✓ | ✗ | 1 | >120,000 | ✗ | ✗ | ✗ |
| [VARSCOT](https://github.com/BauerLab/VARSCOT) (Wilson *et al.* 2019) | N{20} | NGG | ✗ | ✓ | ✗ | ✗ | ✗ | 3 | Any | ✗ | ✓ | ✗ |

In the columns containing motif descriptions or quantifiers, the text “Any” refers to motifs >0 nt in length. Motifs are defined by a string of (possibly ambiguous) nucleotide characters. Repeated characters are followed by braces ({}) containing comma-separated numbers representing the minimum and maximum number of repeats, respectively. If only one number is listed, then the minimum and maximum are equal. If the second number is omitted, then no maximum exists. Motifs are specified when possible, otherwise a number representing the total number of different motifs allowed is given. Any element that could not be evaluated is given “NA”. The asterisk (\*) indicates that this element differs between the web and command line versions. The column “Ambiguous bases” refers to the ability to identify Targets from regions in the genome that contain ambiguous nucleotides. The column “Integrated ranking” means that results are ordered based on a combination of metrics, and not just one algorithm score.

## Table S2 – Plasmids used in this study

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Purpose | Ordering information | Sequence | Source |
| pADH110 | NAT-marked pSNR52 promoter fragment for use in gRNA expression cassette stitching PCR; for use with pADH119 to generate Target-specific gRNA expression cassette. | https://www.addgene.org/90982/ | [GBK](https://media.addgene.org/snapgene-media/v1.6.2-0-g4b4ed87/sequences/05/10/180510/addgene-plasmid-90982-sequence-180510.gbk) | (Nguyen *et al.* 2017) |
| pADH119 | NAT-marked “empty” gRNA construct for use in gRNA expression cassette stitching PCR. Use with pADH110 to generate Target-specific part 2 of 2 for *C. albicans* LEUpOUT CRISPR system. | https://www.addgene.org/90985/ | [GBK](https://media.addgene.org/snapgene-media/v1.6.2-0-g4b4ed87/sequences/05/13/180513/addgene-plasmid-90985-sequence-180513.gbk) | (Nguyen *et al.* 2017) |
| pADH137 | NAT-marked Cas9 expression construct; part 1 of 2 of *C. albicans* LEUpOUT CRISPR system. Use with pADH118-series gRNA expression constructs. | https://www.addgene.org/90986/ | [GBK](https://media.addgene.org/snapgene-media/v1.6.2-0-g4b4ed87/sequences/05/19/180519/addgene-plasmid-90986-sequence-180519.gbk) | (Nguyen *et al.* 2017) |

The identifier is the term used to represent this plasmid in the source manuscript. Each plasmid is available to order from Addgene at the provided web address.

## Table S3 – Oligonucleotides and synthetic DNA fragments used in this study

|  |  |  |  |
| --- | --- | --- | --- |
| Identifier | Sequence (5’à3’) | Purpose | Source |
| AHO 2144 | CTAAGAAGGGAAAAGCACCAC | *ADE2CDS* AmpF | (This study) |
| AHO 2145 | CTCGGTACAATCTTGTCAATGAG | *ADE2CDS* AmpR | (This study) |
| AHO 2137 | GTGGTGGATTGGTATTTCTTTCTGTG | *ADE2CDS* sF | (This study) |
| AHO 2138 | AAGACCCCAAACATTTTGACTCG | *ADE2CDS* sR | (This study) |
| AHO 2142 | CCCCAATGTGTAACAAGTCATCG | *ADE2CDS* +/AoF | (This study) |
| AHO 2139 | CATTGCCTGTCATTGGTGTTCC | *ADE2CDS* +/AoR | (This study) |
| AHO 2140 | CAGAGTTGTGAGGTCTTGGTG | *ADE2CDS* +/AiF | (This study) |
| AHO 2141 | GGCGTATGATGGTAGAGGTAAC | *ADE2CDS* +/AiR | (This study) |
| AHO 2135 | CACCATAACGTTTACTTGTTTAATATGCTATTGATATCTATATTTTTTTCCTATGTGTAGTGCTTGTATATGCGTGTGTGATGAGAATAAGATGAATAGA | *ADE2CDS* Step 1 dDNA F | (This study) |
| AHO 2136 | TCTATTCATCTTATTCTCATCACACACGCATATACAAGCACTACACATAGGAAAAAAATATAGATATCAATAGCATATTAAACAAGTAAACGTTATGGTG | *ADE2CDS* Step 1 dDNA R | (This study) |
| AHO 2134 | CGTAAACTATTTTTAATTTGAACACCAATGACAGGCAATGGTTTTAGAGCTAGAAATAGC | *ADE2CDS* oligo used for stitching spacer to scaffold for +Target | (This study) |
| AHO 2143 | CGTAAACTATTTTTAATTTGCATATACAAGCACTACACATGTTTTAGAGCTAGAAATAGC | *ADE2CDS* oligo used for stitching spacer to scaffold for AddTag Target | (This study) |
| AHO 2168 | CACATTAGTTGCTCAGGTCAC | *EFG1CDS* AmpF | (This study) |
| AHO 2169 | GTCAATGGATTTGGGAGAAGA | *EFG1CDS* AmpR | (This study) |
| AHO 2161 | TTAACCCCTTTGTGTCCCTT | *EFG1CDS* sF | (This study) |
| AHO 2162 | CCCAAATAGTATAAATTCGTTCATGTC | *EFG1CDS* sR | (This study) |
| AHO 2166 | ACCAATCACCCCAAGTTCAG | *EFG1CDS* +/AoF | (This study) |
| AHO 2163 | GCTGTTGTTGTTGTTGTCCT | *EFG1CDS* +/AoR | (This study) |
| AHO 2164 | CCCCCATACCTTCCAATTCTAC | *EFG1CDS* +/AiF | (This study) |
| AHO 2165 | GACACATTACTGCCACCACTG | *EFG1CDS* +/AiR | (This study) |
| AHO 2159 | AACGAATTAAGATTTGTTCTATTTGACTACCAAGAATATAACCCATATTCCTCGTGTACATCACCTTCTGCTTTCTGCCATAAATTCCAAATTAGATTAT | *EFG1CDS* Step 1 dDNA F | (This study) |
| AHO 2160 | ATAATCTAATTTGGAATTTATGGCAGAAAGCAGAAGGTGATGTACACGAGGAATATGGGTTATATTCTTGGTAGTCAAATAGAACAAATCTTAATTCGTT | *EFG1CDS* Step 1 dDNA R | (This study) |
| AHO 2158 | CGTAAACTATTTTTAATTTGTGGTTGGAATTGCCCCACAGGTTTTAGAGCTAGAAATAGC | *EFG1CDS* oligo used for stitching spacer to scaffold for +Target | (This study) |
| AHO 2167 | CGTAAACTATTTTTAATTTGAGCAGAAGGTGATGTACACGGTTTTAGAGCTAGAAATAGC | *EFG1CDS* oligo used for stitching spacer to scaffold for AddTag Target | (This study) |
| AHO 2156 | TATAAATATCAGGTCATAGATCCCTG | *BRG1CDS* AmpF | (This study) |
| AHO 2157 | CTGCTACAGTATTGTTGTTTGAAC | *BRG1CDS* AmpR | (This study) |
| AHO 2149 | TGCAGCTTTTGTACTACATTTGG | *BRG1CDS* sF | (This study) |
| AHO 2150 | CCAGCTCAGGATATAATTTACAGC | *BRG1CDS* sR | (This study) |
| AHO 2154 | GTCATTCATCAACCACCACCA | *BRG1CDS* +/AoF | (This study) |
| AHO 2151 | ACCTCCACTAATGGTTGATCG | *BRG1CDS* +/AoR | (This study) |
| AHO 2152 | CCACCACAACAACCACAATCAG | *BRG1CDS* +/AiF | (This study) |
| AHO 2153 | CGACCGTTCTTCCCTTTTGTC | *BRG1CDS* +/AiR | (This study) |
| AHO 2147 | GTACTACTGTTCATATTTGATATTTCAACGTTATTTCTCCATCCATACTTCTGGCGGTATTCCTGTTGCTTACCCAACCCAAATTCCTTTAATTCGTCAT | *BRG1CDS* Step 1 dDNA F | (This study) |
| AHO 2148 | ATGACGAATTAAAGGAATTTGGGTTGGGTAAGCAACAGGAATACCGCCAGAAGTATGGATGGAGAAATAACGTTGAAATATCAAATATGAACAGTAGTAC | *BRG1CDS* Step 1 dDNA R | (This study) |
| AHO 2146 | CGTAAACTATTTTTAATTTGGGGCTAAGTGACGATGCAGGGTTTTAGAGCTAGAAATAGC | *BRG1CDS* oligo used for stitching spacer to scaffold for +Target | (This study) |
| AHO 2155 | CGTAAACTATTTTTAATTTGAATACCGCCAGAAGTATGGAGTTTTAGAGCTAGAAATAGC | *BRG1CDS* oligo used for stitching spacer to scaffold for AddTag Target | (This study) |
| AHO 2209 | GCATATTTACTTGCTTGCCTG | *ZRT2US* AmpF | (This study) |
| AHO 2214 | TTGACAGGAATATGGAGGGTA | *ZRT2US* AmpR | (This study) |
| AHO 2203 | GAACCAATCCTTCCACATAGC | *ZRT2US* sF | (This study) |
| AHO 2204 | GCTGGGAATTGATAATGAAAGC | *ZRT2US* sR | (This study) |
| AHO 2208 | TATTGGTCGGATTGGGTTAC | *ZRT2US* +/AoF | (This study) |
| AHO 2205 | TTGCGTTTCGGGTATAATCAC | *ZRT2US* +/AoR | (This study) |
| AHO 2206 | GAGAAGAACCATAAAGTCCAAGC | *ZRT2US* +/AiF | (This study) |
| AHO 2207 | CACCTCAAACCACACACTAC | *ZRT2US* +/AiR | (This study) |
| AHO 2201 | CGATATTGTGTAATTTTACATTTGGGCACAGCATAGCCTGATGCCGTCCGGGTCGTACGCTGCAGGTCGACAG | *ZRT2US* Step 1 dDNA F | (This study) |
| AHO 2202 | TGGTGATGGTTTTTATTAGTGGTTACAAAAATGAACAAGAGAAAATTTGCAATACCACTGTCGACCTGCAGCGTAC | *ZRT2US* Step 1 dDNA R | (This study) |
| SynFrag 1 | GCATATTTACTTGCTTGCCTGATATCCTCGACTCATATACTTTGTAAATTACCTGTCACGTGTTTTTGTGAACTCCGATATTGTGTAATTTTACATTTGGGCACAGCATAGCCTGATGCCGTCCGGGT**accctggtagt**TATCACTCAATTTTTTTTTGTTTTTCACTGTTTTTCTGTCTTGTTGTTCCAAATAACCACTAATATTTCTCTTATACTTGACGATTTTTGGTGACCTATTATAGCTGGCAAGTGAAAGTGAATTAATAATATGCATTTTATAAAGTAGGCTTATTCATAAAATAATTAATTATTATTCAATCTCTAATTGATGTTCAGAAAATTTTTGGTTTGATGCCATACAAAGCAAAAAAAAAAATAATACATCAAAAATAGAACAAATGTAACTTTATGGTATTAAATCGTAATCATACACTTACTGAGAAGAACCATAAAGTCCAAGCTTTATAGAAAAAAGGCTAATGTTCTTTAGCATATGGTTTTTTTATGTCCTGGATTAACAACGTCCTTGGACTTAAGTACGTATGAAAGAACTAGCTAATAATTTAAAGCCAAACTGAGTCTTTCAACAACTACAATAGTGATTATACCCGAAACGCAAAATAATAAAAACTAATATTGACAATTGAATTATTCATTATTGGTCGGATTGGGTTACATTCAGATTGAAATCACGGTTGTAATTGCCGAATCTCTTTTTCATTGTTGTTCCATTTGTAACATTACCAGCTAGAAATGTAGTGTGTGGTTTGAGGTGCGTTTAGA**cagctgttcta**TATTGCAAATTTTCTCTTGTTCATTTTTGTAACCACTAATAAAAACCATCACCAATTGACAATGAGTAAAAACTTTAAAAAAAAAGTAAAAATTAGAAAGAAAAAGTCAATCTCCCTTTTGTTGTAATTTATTTATAAATACCCTCCATATTCCTGTCAA | *ZRT2US* Step 2 dDNA for AB01 | (This study) |
| SynFrag 2 | GCATATTTACTTGCTTGCCTGATATCCTCGACTCATATACTTTGTAAATTACCTGTCACGTGTTTTTGTGAACTCCGATATTGTGTAATTTTACATTTGGGCACAGCATAGCCTGATGCCGTCCGGGT**cagggtcgcta**TATCACTCAATTTTTTTTTGTTTTTCACTGTTTTTCTGTCTTGTTGTTCCAAATAACCACTAATATTTCTCTTATACTTGACGATTTTTGGTGACCTATTATAGCTGGCAAGTGAAAGTGAATTAATAATATGCATTTTATAAAGTAGGCTTATTCATAAAATAATTAATTATTATTCAATCTCTAATTGATGTTCAGAAAATTTTTGGTTTGATGCCATACAAAGCAAAAAAAAAAATAATACATCAAAAATAGAACAAATGTAACTTTATGGTATTAAATCGTAATCATACACTTACTGAGAAGAACCATAAAGTCCAAGCTTTATAGAAAAAAGGCTAATGTTCTTTAGCATATGGTTTTTTTATGTCCTGGATTAACAACGTCCTTGGACTTAAGTACGTATGAAAGAACTAGCTAATAATTTAAAGCCAAACTGAGTCTTTCAACAACTACAATAGTGATTATACCCGAAACGCAAAATAATAAAAACTAATATTGACAATTGAATTATTCATTATTGGTCGGATTGGGTTACATTCAGATTGAAATCACGGTTGTAATTGCCGAATCTCTTTTTCATTGTTGTTCCATTTGTAACATTACCAGCTAGAAATGTAGTGTGTGGTTTGAGGTGCGTTTAGA**accttgttggt**TATTGCAAATTTTCTCTTGTTCATTTTTGTAACCACTAATAAAAACCATCACCAATTGACAATGAGTAAAAACTTTAAAAAAAAAGTAAAAATTAGAAAGAAAAAGTCAATCTCCCTTTTGTTGTAATTTATTTATAAATACCCTCCATATTCCTGTCAA | *ZRT2US* Step 2 dDNA for AB10 | (This study) |
| SynFrag 3 | GCATATTTACTTGCTTGCCTGATATCCTCGACTCATATACTTTGTAAATTACCTGTCACGTGTTTTTGTGAACTCCGATATTGTGTAATTTTACATTTGGGCACAGCATAGCCTGATGCCGTCCGGGT**cagggtcgcta**TATCACTCAATTTTTTTTTGTTTTTCACTGTTTTTCTGTCTTGTTGTTCCAAATAACCACTAATATTTCTCTTATACTTGACGATTTTTGGTGACCTATTATAGCTGGCAAGTGAAAGTGAATTAATAATATGCATTTTATAAAGTAGGCTTATTCATAAAATAATTAATTATTATTCAATCTCTAATTGATGTTCAGAAAATTTTTGGTTTGATGCCATACAAAGCAAAAAAAAAAATAATACATCAAAAATAGAACAAATGTAACTTTATGGTATTAAATCGTAATCATACACTTACTGAGAAGAACCATAAAGTCCAAGCTTTATAGAAAAAAGGCTAATGTTCTTTAGCATATGGTTTTTTTATGTCCTGGATTAACAACGTCCTTGGACTTAAGTACGTATGAAAGAACTAGCTAATAATTTAAAGCCAAACTGAGTCTTTCAACAACTACAATAGTGATTATACCCGAAACGCAAAATAATAAAAACTAATATTGACAATTGAATTATTCATTATTGGTCGGATTGGGTTACATTCAGATTGAAATCACGGTTGTAATTGCCGAATCTCTTTTTCATTGTTGTTCCATTTGTAACATTACCAGCTAGAAATGTAGTGTGTGGTTTGAGGTGCGTTTAGA**cagctgttcta**TATTGCAAATTTTCTCTTGTTCATTTTTGTAACCACTAATAAAAACCATCACCAATTGACAATGAGTAAAAACTTTAAAAAAAAAGTAAAAATTAGAAAGAAAAAGTCAATCTCCCTTTTGTTGTAATTTATTTATAAATACCCTCCATATTCCTGTCAA | *ZRT2US* Step 2 dDNA for AB11 | (This study) |
| AHO 2200 | CGTAAACTATTTTTAATTTGAAATTGAGTGATAactaccaGTTTTAGAGCTAGAAATAGC | *ZRT2US* oligo used for stitching spacer to scaffold for +Target | (This study) |
| AHO 2195 | CGTAAACTATTTTTAATTTGCGTACGCTGCAGGTCGACAGGTTTTAGAGCTAGAAATAGC | *ZRT2US* oligo used for stitching spacer to scaffold for AddTag Target | (This study) |
| AHO 2666 | GTCAGTTTCCCATACACATAAGG | *WOR1USd* AmpF | (This study) |
| AHO 2667 | AGCAAGTATAGCCGTCATCT | *WOR1USd* AmpR | (This study) |
| AHO 2661 | CTCTCATCAACAACAACGTCA | *WOR1USd* sF | (This study) |
| AHO 2662 | AATAGTAGACTCCCTAACAGAGC | *WOR1USd* sR | (This study) |
| AHO 2664 | GGAAACTAACCTAACACACAAAC | *WOR1USd* +/AoF | (This study) |
| AHO 2663 | GTTTGTGTGTTAGGTTAGTTTCC | *WOR1USd* +/AoR | (This study) |
| AHO 2664 | GGAAACTAACCTAACACACAAAC | *WOR1USd* +/AiF | (This study) |
| AHO 2665 | TCCCACCCGTCTTTCATAAA | *WOR1USd* +/AiR | (This study) |
| AHO 2659 | TAGGGACATTCAATTCGTCTTGAAAATATTAAAATTGACAAGAAAAACTTATTCGTACGCTGCAGGTCGACAG | *WOR1USd* Step 1 dDNA F | (This study) |
| AHO 2660 | TGGTAGGTTCTGTCATTTATTGCTCTATTTTATAGTATTTAAAGTTTAAACTTTCCACTGTCGACCTGCAGCGTAC | *WOR1USd* Step 1 dDNA R | (This study) |
| AHO 2658 | CGTAAACTATTTTTAATTTGTCATACACCAAGAAAACTCAGTTTTAGAGCTAGAAATAGC | *WOR1USd* oligo used for stitching spacer to scaffold for +Target | (This study) |
| AHO 2195 | CGTAAACTATTTTTAATTTGCGTACGCTGCAGGTCGACAGGTTTTAGAGCTAGAAATAGC | *WOR1USd* oligo used for stitching spacer to scaffold for AddTag Target | (This study) |
| SynFrag 4 | GTCAGTTTCCCATACACATAAGGGAATGACCACACTCAAAAGTAATATCATAACTACAGGGCATAAAGCATATCACCTAGGGACATTCAATTCGTCTTGAAAATATTAAAATTGACAAGAAAAACTTATTCAAAGGGAGACCAAAAATACAGATTACCAACTATGTACACCCTAGAAAGAACTCAAAAAACGTAACCTTCGTTTCAAGTTGCACTTTAAAACAACAAATCCTGCTTTGATCAGATGAAACTATAATGCACGAATATGGAAACTAACCTAACACACAAACAATATATCATACACCAAGAAAACTCATGGTTTGTTGTTGTTGTTAGTGTATAATGTTAAAAAACTCTATTTTCACAATGACCCAAATAAAACCAAAAAAACACTAAGAAG**acccttgcg**TTTGAAACTTTTCAAAATGTATAGAGATCCCAAATCTAAAAAATGTTATTCACTATGGTTGTTGTTGTTTATTCAGAATTTAGTTATGGTTATATTAATGAAACTGTAACATAAAAAAAAACAAGGGAATAATTAGAGTTTTACAAGAAATTTATGAAAGACGGGTGGGAAAAAAGTTTAAACTTTAAATACTATAAAATAGAGCAATAAATGACAGAACCTACCAGTAGTGATTCATAAATTATTATTTCTTGTTATACAATCAAAACCCCAGATATGATAACAGGAAAAAAAAAAGTACTTATATAGATGACGGCTATACTTGCT | *WOR1USd* Step 2 dDNA for AB1 | (This study) |
| AHO 2664 | GGAAACTAACCTAACACACAAAC | *WOR1USp* AmpF | (This study) |
| AHO 2676 | CCCACCTTCTCCCTCTTTC | *WOR1USp* AmpR | (This study) |
| AHO 2666 | GTCAGTTTCCCATACACATAAGG | *WOR1USp* sF | (This study) |
| AHO 2671 | CTCCCCCAACAACAAGTCTT | *WOR1USp* sR | (This study) |
| AHO 2675 | GAGCAATAAATGACAGAACCTACC | *WOR1USp* +/AoF | (This study) |
| AHO 2672 | TCCCACCCGTCTTTCATAA | *WOR1USp* +/AoR | (This study) |
| AHO 2673 | CACTATGGTTGTTGTTGTTTATTCAG | *WOR1USp* +/AiF | (This study) |
| AHO 2674 | AGTAGACTCCCTAACAGAGC | *WOR1USp* +/AiR | (This study) |
| AHO 2669 | ACAATGACCCAAATAAAACCAAAAAAACACTAAGAAGTTAAACTTTTTTGAAACCACTGTCGACCTGCAGCGTAC | *WOR1USp* Step 1 dDNA F | (This study) |
| AHO 2670 | ATTTTTGCATGTTCTATTTTTAGTCCATACATAATGTAACGCACACACATTAGACGTACGCTGCAGGTCGACAG | *WOR1USp* Step 1 dDNA R | (This study) |
| AHO 2668 | CGTAAACTATTTTTAATTTGATTTATGAATCACTACTGGTGTTTTAGAGCTAGAAATAGC | *WOR1USp* oligo used for stitching spacer to scaffold for +Target | (This study) |
| AHO 2195 | CGTAAACTATTTTTAATTTGCGTACGCTGCAGGTCGACAGGTTTTAGAGCTAGAAATAGC | *WOR1USp* oligo used for stitching spacer to scaffold for AddTag Target | (This study) |
| SynFrag 5 | GGAAACTAACCTAACACACAAACAATATATCATACACCAAGAAAACTCATGGTTTGTTGTTGTTGTTAGTGTATAATGTTAAAAAACTCTATTTTCACAATGACCCAAATAAAACCAAAAAAACACTAAGAAGTTAAACTTTTTTGAAACTTTTCAAAATGTATAGAGATCCCAAATCTAAAAAATGTTATTCACTATGGTTGTTGTTGTTTATTCAGAATTTAGTTATGGTTATATTAATGAAACTGTAACATAAAAAAAAACAAGGGAATAATTAGAGTTTTACAAGAAATTTATGAAAGACGGGTGGGAAA**cgcaaacccttgcg**AAATACTATAAAATAGAGCAATAAATGACAGAACCTACCAGTAGTGATTCATAAATTATTATTTCTTGTTATACAATCAAAACCCCAGATATGATAACAGGAAAAAAAAAAGTACTTATATAGATGACGGCTATACTTGCTCAAGTGAGTTTGATGTGATTTTTAACACGCTCTGTTAGGGAGTCTACTATTTTTTTTTCTGGCGATAACAATAAGAAATCTCTAATGTGTGTGCGTTACATTATGTATGGACTAAAAATAGAACATGCAAAAATTGCGAGAAAGAAAGCGAGTGAGTAAGGGCGTGCGTGCGTGCATGAGTGAAAGAGGGAGAAGGTGGG | *WOR1USp* Step 2 dDNA for AB1 | (This study) |
| AHO 2686 | ACACACACACACACAATCACAC | *WOR2DS* AmpF | (This study) |
| AHO 2687 | TAGCAAGGCAACCATCAAGC | *WOR2DS* AmpR | (This study) |
| AHO 2680 | CTACTACTGATGGTCTACTGATGG | *WOR2DS* sF | (This study) |
| AHO 2681 | CGTTTGTAGATGGTTCTGGTTTG | *WOR2DS* sR | (This study) |
| AHO 2685 | ATCGCTCCTTGTGTTTGTGTG | *WOR2DS* +/AoF | (This study) |
| AHO 2682 | CCCACACAAACACAAGGAGC | *WOR2DS* +/AoR | (This study) |
| AHO 2683 | TTGTGGAAGTGTAAGAGGGA | *WOR2DS* +/AiF | (This study) |
| AHO 2684 | CTGCTTGCTAAACCCAAACC | *WOR2DS* +/AiR | (This study) |
| AHO 2678 | CTAAAAACCAACAAGTTACTTGATAGAACCTCGATTTCATTATGAATTCCACACGTACGCTGCAGGTCGACAG | *WOR2DS* Step 1 dDNA F | (This study) |
| AHO 2679 | AATACAAATACAGATGACACCAAAAAGAAAAAAGTTAAACTTGTAATAGTTAATCCACTGTCGACCTGCAGCGTAC | *WOR2DS* Step 1 dDNA R | (This study) |
| AHO 2677 | CGTAAACTATTTTTAATTTGATCGCTCCTTGTGTTTGTGTGTTTTAGAGCTAGAAATAGC | *WOR2DS* oligo used for stitching spacer to scaffold for +Target | (This study) |
| AHO 2195 | CGTAAACTATTTTTAATTTGCGTACGCTGCAGGTCGACAGGTTTTAGAGCTAGAAATAGC | *WOR2DS* oligo used for stitching spacer to scaffold for AddTag Target | (This study) |
| SynFrag 6 | ACACACACACACACAATCACACAAAATTAGCACTACTAAATGTTTGAGAATGATTCGAATCAAGGGAAACTAAAAACCAACAAGTTACTTGATAGAACCTCGATTTCATTATGAATTCCACATACACAATAATACAGTACCAAAAGTTTAAATTTAAAAAAAAAATCAGCCCATTAGAGAAATCTAGATGTAGATATATTTTGTGGAAGTGTAAGAGGGATAAGCCATTTGTAATTTTACACAATTAATCGCTCCTTGTGTTTGTGTGGGAAAAACTTTGCAATTGGTTGATTGTGCAACAATTGCTAAAACATTGGTTACCCATTTTCCTTTTTTTGCAATTTCCAAATAATAATAATGATAATACTTATCAAAACAAAGAAACAATTAACGAGACAAGTTTAAATCAAACTCAATACAATTCATAAACTCTAACTGG**cgcaagggt**GTTTTCTATTTTTTGTTTGTGAATGTATTACAATAAATTGAATTTTGATCGAAATATTAATCGGGGCTAGAGTGTGGTTTGGGTTTAGCAAGCAGCTATTGTTTGAAAAAAATTAAAATGACTGCATTAACTATTACAAGTTTAACTTTTTTCTTTTTGGTGTCATCTGTATTTGTATTTATTGCATGGGAAAGACAATACAGTAGTAATAACGAAACTATCAACCACGAAAAGAGGAAATATCCCTCAACTTTCCAAATTTAATTCAAAAGATACTAAAAAAAACCTTGAGTCAACAATAGAATTTATTGAAACTTAATTCTCCTCATGTGGATTCTTTATTTGCTTGATGGTTGCCTTGCTA | *WOR2DS* Step 2 dDNA for AB1 | (This study) |
| AHO 1096 | GACGGCACGGCCACGCGTTTAAACCGCC | gRNA cassette part 1 F | (Nguyen *et al.* 2017) |
| AHO 1098 | CAAATTAAAAATAGTTTACGCAAG | gRNA cassette part 1 R | (Nguyen *et al.* 2017) |
| AHO 1099 | GTTTTAGAGCTAGAAATAGCAAGTT | gRNA cassette part 2 F | (Nguyen *et al.* 2017) |
| AHO 1097 | CCCGCCAGGCGCTGGGGTTTAAACACCG | gRNA cassette part 2 R | (Nguyen *et al.* 2017) |
| AHO 1237 | AGGTGATGCTGAAGCTATTGAAG | gRNA full cassette F | (Nguyen *et al.* 2017) |
| AHO 1238 | TGTATTTTGTTTTAAAATTTTAGTGACTGTTTC | gRNA full cassette R | (Nguyen *et al.* 2017) |

Shared upstream and downstream primers sF and sR are colored blue. Sequences homologous to the AmpF and AmpR primers, used to amplify the Step 2 dDNA from wild-type gDNA or synthetic DNA templates, are colored green. Nucleotides that are homologous to the AddTag-selected spacer sequences in the first and second round of genome editing are colored red, and fuchsia, respectively, with the associated PAM sequences in brick and violet. In first round donor DNA sequences (Step 1 dDNA), the upstream and downstream homology regions are given yellow and orange backgrounds, and any addtag insert sequence is given a pink background. **Bold**, lower-case letters are nucleotides that encode for the Zap1 and Wor1 binding sites. Sequences are listed in the 5’ to 3’ orientation. Oligonucleotide sequences used in this study did not require any special modifications or purifications. SynFrag sequences used to generate *ZRT2 US*, *WOR1USd*, *WOR1USp*, and *WOR2DS* dDNAs were synthesized as dsDNA fragments.

## Table S4 – Strains used in this study

|  |  |  |  |
| --- | --- | --- | --- |
| Identifier | Strains | Genotype | Source |
| *ADE2CDS* +/+ | AHY 940 |  | (Nguyen *et al.* 2017) |
| *ade2CDS* ∆/∆ | AHY 1338,  AHY 1347 |  | (This study) |
| *ADE2CDS* AB/AB | AHY 1267,  AHY 1268 |  | (This study) |
| *BRG1CDS* +/+ | AHY 940 |  | (Nguyen *et al.* 2017) |
| *brg1CDS* ∆/∆ | AHY 1219,  AHY 1220 |  | (This study) |
| *BRG1CDS* AB/AB | AHY 1263,  AHY 1264 |  | (This study) |
| *EFG1CDS* +/+ | AHY 940 |  | (Nguyen *et al.* 2017) |
| *efg1CDS* ∆/∆ | AHY 1336,  AHY 1337 |  | (This study) |
| *EFG1CDS* AB/AB | AHY 1259,  AHY 1260 |  | (This study) |
| *ZRT2US* +/+ | AHY 940 |  | (Nguyen *et al.* 2017) |
| *zrt2US* ∆/∆ | AHY 1221,  AHY 1222 |  | (This study) |
| *ZRT2US* AB00/AB00 | AHY 1261,  AHY 1262 |  | (This study) |
| *ZRT2US* AB01/AB01 | AHY 1271,  AHY 1272 |  | (This study) |
| *ZRT2US* AB10/AB10 | AHY 1273,  AHY 1274 |  | (This study) |
| *ZRT2US*AB11/AB11 | AHY 1295,  AHY 1296 |  | (This study) |
| *WOR1USd* +/+ | AHY 940 |  | (Nguyen *et al.* 2017) |
| *wor1USd* ∆/∆ | AHY 1447,  AHY 1448 |  | (This study) |
| *WOR1USd* AB0/AB0 | AHY 1449,  AHY 1450 |  | (This study) |
| *WOR1USd* AB1/AB1 | AHY 1451,  AHY 1452 |  | (This study) |
| *WOR1USp* +/+ | AHY 940 |  | (Nguyen *et al.* 2017) |
| *wor1USp* ∆/∆ | AHY 1453,  AHY 1454 |  | (This study) |
| *WOR1USp* AB0/AB0 | AHY 1455,  AHY 1456 |  | (This study) |
| *WOR1USp* AB1/AB1 | AHY 1457,  AHY 1458 |  | (This study) |
| *WOR2DS* +/+ | AHY 940 |  | (Nguyen *et al.* 2017) |
| *WOR2DS* ∆/∆ | AHY 1459,  AHY 1460 |  | (This study) |
| *WOR2DS* AB0/AB0 | AHY 1461,  AHY 1462 |  | (This study) |
| *WOR2DS* AB1/AB1 | AHY 1463,  AHY 1464 |  | (This study) |

The identifier is the term used to represent this strain in the manuscript. Each strain name listed for an identifier represents a wholly independent biological derivation of the strain. Genotypes are listed through the standard *Candida* syntax: homologous chromosomes are separated by horizontal lines; linked loci are co-located in the same word, and non-linked loci are separated by white space; loci are represented by the text of their wild-type gene name; wild-type genes are capitalized, and mutant genes are lower-case; and specific modifications to genes at a locus begin with “::”. Superscript 0 and 1 numbers represent allelic forms, with 0 being unmodified from wild-type, and 1 being modified. Subscript CDS indicates the Feature of interest is only a portion of the full chromosomal gene, and subscript US represents a section of the upstream intergenic region of the gene. The sequence orientations for mintag and addtag inserts are indicated with the start as lower contig position and the end as higher contig position, as they appear in the reference genome (i.e. “+” orientation relative to the contig).

## Table S5 – Summary of polymorphisms in dDNA sequences

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Polymorphisms/Length (nt) | | | | | | | | |
|  |  | Step 1 dDNA | | | eFeature | | | Step 2 dDNA | | |
| Gene | Step 1 dDNA type | US | Insert | DS | US | +Feature | DS | US | ✲Feature | DS |
| *ADE2CDS* | mintag | 0/49 | CC | 0/49 | 0/0 | 0/1707 | 0/0 | 0/133 | AB: 0/1707 | 0/140 |
| *BRG1CDS* | mintag | 0/50 | - | 0/50 | 3/42 | 13/1269,1272 | 0/0 | 3/221 | AB: ND | 1/115 |
| *EFG1CDS* | mintag | 0/49 | CCTC | 0/47 | 0/1 | 6/1658,1653 | 1/12 | 1/149 | AB: ND | 2/170 |
| *ZRT2US* | addtag | 0/53 | CGTACGCTGCAGGTCGACAGTGG | 0/54 | 0/0 | 5/667,666 | 0/0 | 0/128 | AB00: 0/667  AB01: 0/667  AB10: 0/667  AB11: 0/667 | 2/160 |
| *WOR1USd* | addtag | 0/53 | CGTACGCTGCAGGTCGACAGTGG | 0/54 | 0/269 | 0/9 | 0/172 | 0/130 | AB0: 0/450  AB1: 0/450 | 0/155 |
| *WOR1USp* | addtag | 0/53 | CCACTGTCGACCTGCAGCGTACG | 0/54 | 0/165 | 0/14 | 0/221 | 0/149 | AB0: 0/400  AB1: 0/400 | 0/120 |
| *WOR2DS* | addtag | 0/53 | CGTACGCTGCAGGTCGACAGTGG | 0/54 | 0/317 | 0/9 | 0/125 | 0/122 | AB0: 0/451  AB1: 0/451 | 0/239 |

The “US”, “DS”, “Insert”, and “Feature” columns, display an “a/b” format, with “a” representing the number of polymorphisms within “b” contiguous nucleotides. The +Feature columns represent the full input Feature that was expanded to circumvent polymorphisms, with the “US” and “DS” columns representing the lengths of Feature expansion. For the Step 1 dDNA and Step 2 dDNA columns, the “US” and “DS” columns represent the flanking homology arms. Please note that the Step 1 dDNAs do not contain polymorphisms, but the Step 2 dDNAs do. Some genomic regions have alleles with different lengths, which are shown as comma-separated list of lengths. The “ND” indicates that polymorphisms in restored loci were not determined.

# Supplemental materials and methods

## “All-in-1” gRNA cassette stitching

This protocol describes the “All-in-1” gRNA expression cassette stitching method, an adaptation of the cloning-free two-step assembly method previously described by Nguyen et al (Nguyen *et al.* 2017). The All-in-1 approach assembles two universal gRNA expression cassette fragments (Fragments A and B) into an intact Target-specific gRNA expression cassette in a single reaction using a custom bridging gRNA oligo and conserved amplification primers (Figure S4). While the original method requires the generation of a new Fragment B for each unique gRNA target, the All-in-1 simply requires a single unique oligonucleotide and two universal PCR fragments as templates. The All-in-1 approach cuts the time to generate gRNA expression cassettes nearly in half as compared to the traditional method and is as just as efficient in creating gene knockouts (unpublished).

1. PCR amplify the universal “A” fragment from gRNA plasmid 1of2

* PCR Mix (makes enough “A” fragment for >75 “C” fragment stitching PCRs):

1. 75.5µL H2O
2. 20µL 5x Phusion HF buffer
3. 2µL dNTP mix (10 mM each dNTP)
4. 1µL pADH110 (1ng/µL)
5. 0.5µL AHO1096 (100uM)
6. 0.5µL AHO1098 (100uM)
7. 0.5µL Phusion polymerase

* “A” fragment PCR cycling conditions:

1. 98˚C, 30sec
2. 98˚C, 20sec
3. 58˚C, 20sec
4. 72˚C, 30sec
5. Return to step 2 for a total of 30 cycles
6. End
7. PCR amplify the universal “B” fragment from gRNA plasmid 2of2

* PCR Mix (makes enough “B” fragment for >75 “C” fragment stitching PCRs):
  1. 75.5µL H2O
  2. 20µL 5x Phusion HF buffer
  3. 2µL dNTP mix (10 mM each dNTP)
  4. 1µL pADH119 (1ng/µL)
  5. 0.5µL AHO1097 (100uM)
  6. 0.5µL AHO1099 (100uM)
  7. 0.5µL Phusion polymerase
* “B” fragment touchdown PCR cycling conditions:

1. 98˚C, 30sec
2. 98˚C, 20sec
3. 65˚C, 20sec
4. 72˚C, 30sec
5. Return to step 2 for a total of 10 cycles, reducing annealing temperature by 1˚C each cycle.
6. 98˚C, 20sec
7. 55˚C, 20sec
8. 72˚C, 30sec
9. Return to step 6 for a total of 25 cycles
10. End
11. All-in-1 stitching PCR to amplify unique “C” fragment gRNA expression cassette

* PCR Mix (makes enough “C” fragment for two transformations):

1. 73.5µL water
2. 20µL 5xHF buffer
3. 2µL dNTPs 10mM
4. 1µL universal A Fragment (See note 1 below)
5. 1µL universal B Fragment (See note 1 below)
6. 0.5µL AHO1237 100µM
7. 0.5µL AHO1238 100µM
8. 1 µL custom gRNA oligo 100nM (See note 2 below)
9. 0.5 µL Phusion polymerase

* “C” fragment touchdown PCR cycling conditions:
  1. 98˚C, 30sec
  2. 98˚C, 15sec
  3. 60˚C, 15sec
  4. 72˚C, 60sec
  5. Return to step 2 for a total of 5 cycles, reducing annealing temperature by 1˚C each cycle.
  6. 98˚C, 15sec
  7. 66˚C, 15sec
  8. 72˚C, 60sec
  9. Return to step 6 for a total of 30 cycles
  10. End

Notes:

1. The A and B fragment PCR products can be added directly to the C fragment PCR reaction without any post-PCR purification.
2. It is critical that the custom gRNA oligo is at low concentrations so that the B fragment does not take over the PCR reaction.

# Supplemental results

## *ZRT2* strains

We assessed growth of the modified *ZRT2* strains on zinc-sufficient and zinc-deficient media (Figure S11) and observed subtle yet consistent differences in growth between each genotype that were comparable under both growth conditions, implying that changes in environmental zinc levels may not influence Zap1-dependent regulation of *ZRT2*. Under both growth conditions, the *ZRT2US* AB00/AB00 and *ZRT2US* AB10/AB10 strains exhibited growth that was indistinguishable from the wild-type *ZRT2*US +/+ strain, while the *zrt2US* ∆/∆, *ZRT2US* AB01/AB01, and *ZRT2US* AB11/AB11 strains showed slightly reduced growth relative to the wild-type reference strain. This suggests that the CDS-proximal Zap1 binding site upstream of *ZRT2* is predominantly responsible for Zap1-mediated induction of *ZRT2*, and that the CDS-distal Zap1 binding site may not play a significant role under the growth conditions tested. Interestingly, disruption of the CDS-proximal Zap1 binding site upstream of *ZRT2* resulted in a similar growth defect as complete deletion of the entire region that encompasses both Zap1 binding sites, again highlighting the importance of the CDS-proximal Zap1 binding site. We note that although the observed growth defects are relatively subtle, this is not entirely unexpected. *C. albicans* has several genes that work in concert to maintain homeostasis of zinc levels. One explanation for why *ZRT2US* +/+, *ZRT2US* AB00/AB00, and *ZRT2US* AB10/AB10 show similar growth phenotypes is that other zinc transporters, such as *ZRT1* are compensating for any deficiencies to *ZRT2* (Crawford *et al.* 2018).

# Software overview

## Target selection

AddTag automates the process of identifying quality Targets that make RGN-induced DSBs within, or adjacent to, the user-defined genomic Feature to be edited. AddTag searches both strands of the input or generated DNA for the input RGN Target motif, and does basic evaluations of the %GC and Poly-T count. To further evaluate potential gRNA sequences, we have implemented the most commonly utilized calculations designed for this purpose—Azimuth (Doench *et al.* 2016) and DeepCpf1 (Kim *et al.* 2016; Kim *et al.* 2018) for on-target scores; CFD (Doench *et al.* 2016), Hsu-Zhang (Hsu *et al.* 2013), and a general-purpose linear model for off-target scores; and Doench-2014 (Doench *et al.* 2014), CRISPRater (Labuhn *et al.* 2017), Moreno-Mateos (Moreno-Mateos *et al.* 2015) and Housden (Housden *et al.* 2015b) for multi-purpose scores. However, unlike most utilities for evaluating RGN Targets, AddTag contains a holistic method for ranking Targets based on the scores resulting from each of these Algorithms (Figure S7, Figure S8, Figure S14) (Supplement – Scoring and ranking Targets). AddTag implements more scoring Algorithms than most other software package we compared it to (Table S1), providing a level of flexibility and robustness in gRNA design that is unmatched. AddTag establishes a unified framework for working with both 3’-adjacent and 5’-adjacent PAM sequences, and both blunt and staggered cuts, such as with Cas9 and Cas12a. The AddTag software includes the most extensive compendium of all identified RGN Target motifs (Xu *et al.* 2015; Leenay and Beisel 2017; Raikwar *et al.* 2019), which is accessible on the command line. It describes the characteristics for each RGN that have been published, including the highest-efficiency on-target Target motif, the lengths of the spacers tested, the empirical biological system, and any restriction positions determined.

## Feature expansion

A powerful element of the 2-step genome editing approach facilitated by AddTag is that users need not be concerned about whether their genomic Feature contains a unique Target sequence for their RGN of choice. Users can simply define the Feature that they are interested in modifying and provide the sequence to which they wish to change the Feature, then AddTag will design all the dDNA, primer, and gRNA sequences necessary to achieve this goal. At the heart of this capability is a powerful Feature selection and expansion utility which expands a user-defined Feature to enable editing of a Feature, via 2-step editing, that cannot be efficiently edited via direct (1-step) editing (Figure S15). Even if the Feature does contain a quality RGN Target sequence, expansion may still be needed if the flanking homology regions contain allelic polymorphisms that surpass user-defined thresholds. To expand a feature, AddTag creates an “expanded Feature” by incorporating additional upstream and/or downstream sequences until the criteria for RGN Target quality and flanking homology are met. Then this expanded Feature is used to determine the necessary dDNA sequences for deletion and restoration, or modification, of the original user-defined Feature. The Feature expansion utility is highly configurable with many user configurable options (e.g., maximum Feature size, directionality of expansion, or constraints that prevent expansion into neighboring annotations).

## Donor DNA generation

A critical element of successful RGN-mediated genome editing is an effectively designed dDNA sequence. AddTag automatically designs dDNA sequences for each genome editing step, and designs PCR primers for amplifying Step 2 dDNA from a wild-type genome template. Each dDNA sequence has three elements in its basic structure: a region of homology to the gDNA upstream of the Feature, the insert (addtag), and a region of homology to the gDNA downstream of the Feature. Users can specify whether dDNA homology arms should avoid or require polymorphisms among input homologous Features. If the dDNA design for a given Feature fails to pass these criteria, then the Feature will be expanded as described above. The AddTag software gives several options for addtags in the Step 1 dDNA (Figure S6). AddTag software can construct Step 1 dDNAs with unique (addtag) or identical (unitag), full-length AddTag Targets, so experimental loci can be edited in isolation or in parallel. AddTag can construct Step 1 dDNAs with minimal extrinsic DNA (mintag) while also ensuring a high-quality AddTag Target for efficient second step RGN restriction. AddTag also supports traditional (1-step) genome editing using any user-defined dDNA insert (Figure 1A). Following dDNA generation, AddTag performs *in silico* recombination to determine if dDNA homology arms possess significant similarity to non-target locations in the gDNA (Figure S10).

## Automated primer design

AddTag designs an integrated minimal set of PCR primers with which users can amplify dDNA fragments and verify genome edits throughout 1- or 2-step genome editing applications. The AmpF and AmpR primers are selected to efficiently amplify Step 2 dDNA fragments for native locus add-backs (Figure 4A). The remaining primers are designed to produce unique PCR products that are indicative of successful deletion and subsequent restoration or modification of the Feature being edited (Figure 4E). For example, when a Feature is deleted by being replaced by an addtag sequence, amplifying ∆gDNA with the sF and sR primers yields a positive PCR product of a size consistent with the removal of the original Feature from +gDNA. Upon successful reintegration of the wild-type Feature, or integration of a modified version, then the sF and +oR , or sF and AoR primer pairs, respectively, yields a positive PCR product that is indicative of the intended genomic edit. Since the dDNA integration verification PCR primer pairs, such as sF and AoR, are designed to amplify across the regions of flanking homology that are used to integrate the dDNA fragments at each step of genome editing, a positive PCR product is indicative of the integration of the dDNA fragment at the intended locus (Figure 4C, Figure 4E).

The PCR primer design utility selects primers based on user-defined allelic specificity (Figure S16); melting temperature; %GC; length; propensity to form hairpins, homodimers, and heterodimers, and many more metrics (Figure S9) (Supplement – Evaluating Primer suitability). In addition to ensuring that each individual primer meets or exceeds all of the evaluation metrics, the primer design utility further refines the selected primers to ensure that the full complement of primers for any given 1- or 2-step genome editing application are optimized to work under identical PCR conditions. Because all of the selected primers are co-optimized (Figure 4D), any of the primer pair combinations used in genotype verification (Figure 4E) can be run in parallel under identical PCR reactions on the same thermal cycler, thus reducing the need for fragment-specific PCR optimization and enabling higher throughput. Although this PCR primer co-optimization step significantly increases the processor time needed to run the AddTag software, we believe that the resulting gains in efficiency and reduced complexity during the genotype verification steps far outweigh the added computational cost. We note that all strains generated in this study were constructed using target sites, dDNAs, and PCR primers that were exclusively designed by the automated AddTag software. In all cases, the desired genome edits were easily obtained, and the expected banding patterns were observed from PCR-based genotype verifications (for a representative example see Figure S3).

## Comparison to previous software

Many software packages exist for facilitating CRISPR/Cas-based genome edits with varying versatility (Ding *et al.* 2016; Hanna and Doench 2020) (Table S1). Some provide interactive graphical interfaces (Oliveros *et al.* 2016), but require user intervention at multiple steps in the design process (Hough *et al.* 2017). Some CRISPR/Cas design programs are tailored to specific sets of genomes (Table S1, “Genomes”), and pre-computed enzyme binding computations (Hodgkins *et al.* 2015; Park *et al.* 2016; Rauscher *et al.* 2016; Perez *et al.* 2017). The programs with the simplest outputs order candidate Targets based on a single scoring algorithm (Table S1, “Integrated ranking”). Many let users choose a Target scoring algorithm and PAM compatible with their biological system and chosen RGN (Montague *et al.* 2014; Park *et al.* 2015; Haeussler *et al.* 2016; Labun *et al.* 2016). Existing tools simultaneously calculate multiple scoring algorithms (Montague *et al.* 2014; Haeussler *et al.* 2016; Labun *et al.* 2016; Hough *et al.* 2017). CRISPR4P (Rodríguez-López *et al.* 2017) introduced limited forms of automated knock-in dDNA and cPCR primer design. VARSCOT (Wilson *et al.* 2019) introduced ambiguous nucleotide compatibility and allows for uncertainty in genome coverage and ploidy. CRISPRdirect (Naito *et al.* 2014) introduced integrated Target ranking.

We developed AddTag with the goal of maximizing utility through computational flexibility without requiring continual user input. AddTag incorporates many of the previously described software features into a single tool (Table S1): any genome can be used, and any arbitrary Target motif (spacer constraints, cutting arrangement, and PAM sequence) can be used. This makes AddTag ideal for labs that have a high-volume of genome editing demands, labs that edit non-model species or species with significant divergence from reference genomes, and labs that are using novel RGNs. AddTag is a generalized approach for developing 2-step CRISPR/Cas genome editing experiments. After each step, the genomes are assayed for CRISPR/Cas-induced recombination events with a PCR-based assay (Kim and Smithies 1988). Most software packages that choose gRNA Targets and design verification primers are confined to a single editing step and genome (Rodríguez-López *et al.* 2017), but AddTag primer design can span any number of serial genome editing steps, and is applicable to any sequenced genome.

One feature that distinguishes AddTag from other Target identification software is that AddTag de-couples the PAM sequence from the score evaluation. For instance, the typical *in silico* Cas9 gRNA design uses NGG for searching (on-targets), and NRG for scoring (off-targets). Few gRNA design software programs provide the functionality to search for any PAM motif (Table S1), largely because scoring algorithms have not been empirically verified for these. However, AddTag can identify useful gRNA Target sites for any set of arbitrary PAM sequences by adapting existing algorithms. Thus, experimenters can take into account the known flexibility of their chosen RNA-guided nucleases, or use the software without modification to predict binding sites of new RGNs. Because AddTag can consider any number of Target motifs at the same time, researchers can identify which RGNs would be most useful to edit the genome of their chosen biological system.

## Identifying targets and Feature expansion

The objective is to identify gDNA targets within or near to specific genomic Features that can be cut with an experimenter’s RGN of choice, and then produce dDNAs that will replace the Features. If a Feature does not contain a Target, or the user wishes to attempt to find a higher scoring one, the user can direct AddTag to expand the Feature until it finds a suitable +Target. The bounds of the Feature will be spread up to a defined number of nucleotides in both the up- and down-steam directions. All +Targets identified within each expanded Feature (eFeature) derived from the input Feature will be scored. This will generate a 2-step genome editing design in which the first-round ΔdDNA excises more than just the input Feature from the genome, and the second-round Step 2 dDNA re-introduces the extemporaneously-subtracted DNA, along with any intended Insert or modification.

AddTag allows the user to preferentially design homology arms and Targets against polymorphic sites. If the researcher wishes to target multiple Features with the same dDNAs, a list of homologous loci can be provided. AddTag will determine if the adjacent DNA (up- and down-stream, regions flanking a Feature) contain polymorphisms when flanking regions of all homologous loci of the Feature are aligned. The user can direct AddTag to use this information in one of three ways. AddTag can require each homologous locus to have a distinct nucleotide sequence, thereby creating allele-specific dDNAs. AddTag can require all homologous loci to have identical (or a maximal level of polymorphism within the) up- and down-stream flanking regions, thereby creating multi-allelic dDNAs. Finally, AddTag can forgo any alignment of flanking regions, thereby creating allele-agnostic dDNAs. If the homology arms of putative dDNAs would have undesired polymorphism levels, then the Feature can be expanded further to delimit new flanking regions. The expanded feature can take one of several formats (centered on Target, centered on Feature, etc) (Figure S15). After identification, each candidate Target must then pass through a pre-alignment filter (prefilter).

AddTag identifies places on the genome that can be restricted by RGNs by using a Target motif. The Target motif is a string of characters that models the template sequence on the genome, and not the gRNA sequence that the RGN uses. The Target motif thus describes the effective spacer, homologous to the genomic DNA when experimental conditions are met (e.g. temperature and pH), and not the full-length gRNA sequence that may include complex secondary structures where only a portion of the sequence is homologous to the genome (Kocak *et al.* 2019). Candidate Target sites are identified within Features using regular expression-like syntax. The user must include one or more Target motifs specific to the RGN proteins being used. This option takes a string of characters as input, written in the 5’à3’ direction. The greater than “>” or less than “<” characters always point toward the PAM. Thus, Targets with 5’-adjacent PAM sequences like Cas12a would encode this information with the “<” character, and Targets with 3’-adjacent PAM sequences like Cas9 would use the “>” character. The vertical bar “|” represents a double-strand cut. A slash “/” represents a forward strand (sense) cut. Backslashes “\” represent reverse strand (antisense) cuts. Full nucleotide ambiguity codes specified by the IUBMB/IUPAC are supported (Cornish-Bowden 1985). Open “{“ and close “}” braces surrounding a number or a comma-separated pair of numbers represent quantifiers. A period “.” represents a base used for positional information, but not enzymatic recognition. AddTag affords two options that use this syntax: motifs to design gRNA targets against, and motifs to include for off-target calculations only. Any number of motifs can be specified for each affordance.

A typical genome editing experiment using Cas9 (derived from *Streptococcus pyogenes*) would use the following Target motif:

--motifs 'N{17}|N{3}>NGG'

AddTag has the flexibility to deal with the PAM upstream of the spacer, such as the Cas12a (derived from *Acidaminococcus*/*Lachnospiraceae* species) (Zetsche *et al.* 2015):

--motifs 'TTTN<N{19}/.{4}\'

Native microbial gRNA components typically encode for spacer sequences >30 nt in length (Horvath *et al.* 2008; Barrangou and Marraffini 2014). However, early experiments found that shortening the spacer length can increase specificity without severely impacting efficiency. Precedent has set 20 nt length spacer as the standard, but shorter 17-19 nt lengths can be used just as effectively (Fu *et al.* 2014). For ease of use, a list of the most commonly used Target motifs identified across the entire family of Cas RGN molecules is included (such as the one provided in (Swarts and Jinek 2018), but with a deeper sampling of the literature), and can be accessed through the command:

addtag motifs

## Scoring and ranking targets

### Prefilter

In order to score how efficient a spacer sequence is at directing gDNA cutting, Targets identified within Features or Inserts are filtered in two steps. The Prefilter step checks the quality of candidate Target sequences before aligning to the genome and potential dDNA sequences. This reduces the total number of Targets that need to be aligned to the genome and then evaluated by scoring Algorithms. AddTag implements the following Prefilters:

* Upper/lower case masking (ignore, upper-only, lower-only, mixed-lower, mixed-upper, mixed-only). Users can apply several masks to the genome, which can prevent selecting Targets from these masked regions.
* Process ambiguous characters (discard, keep, disambiguate, exclusive). Any ambiguous characters can be equivalently masked. If the user intends to target a region containing ambiguous characters, then AddTag will optionally disambiguate potential spacers.
* Target motif sanity check. Some potential gRNAs derived from ambiguous character expansion may violate the initial motif. This filter ensures that none of these enter into downstream calculations.
* Maximum consecutive T residues. Sequences containing consecutive T residues may cause polymerase termination (Braglia *et al.* 2005).
* Upper/lower %GC content thresholds. The GC content of Cas targets may affects binding specificity (Lin *et al.* 2014).
* Proximal G, which evaluates if the single nucleotide of the spacer adjacent to the PAM is a guanine.

Users can create any number of additional prefilters by subclassing Algorithm in the source/algorithms subdirectory, and setting its prefilter attribute to True.

### Alignment

All potential Target sequences are aligned to the intended gDNA and dDNA sequences. AddTag includes wrappers for the following programs, with all parameters preset to appropriate values: NCBI BLAST+ (Camacho *et al.* 2009), Bowtie 2 (Langmead and Salzberg 2012), BWA (Li and Durbin 2009), and Cas-OFFinder (Bae *et al.* 2014). For the exact shell commands, please refer to either the AddTag source code or an output log file after running AddTag.

### Postfilter

After aligning Target queries to the gDNA and dDNA sequences, the postfilter checks the quality of alignment matches. Each alignment match is scored by all selected Algorithms. Each Algorithm is given a minimum and maximum cutoff value. If the Algorithm score for that match is outside these bounds, then it fails the postfilter. For each Algorithm designated as a postfilter, if the match passes the postfilter criteria, then it is included as a potential on/off-target. By default, PAM-identity and the number of substitutions, insertions, deletions, and errors (Needleman and Wunsch 1970) are included in the postfilter.

### General algorithm interface

Because AddTag uses a variety of scoring algorithms (Algorithm) for evaluating Target suitability, it implements a flexible computational interface for dealing with arbitrary Algorithm requirements. We distinguish the Algorithm type by whether it requires 1 or 2 sequences to be input ( SingleSequenceAlgorithm and PairedSequenceAlgorithm in the source code). 1-sequence Algorithms compare the candidate Target sequence to a model trained on empirical gRNA experiments. 2-sequence algorithms directly compare a gRNA Spacer sequence to a Target sequence, or compare two competing Target sequences.

Any 1-sequence Algorithm can be used to calculate an on-target score. Additionally, several authors claim their Algorithms are appropriate to use for off-target scoring as well. Any 2-sequence Algorithm may be used to calculate off-target scores. After candidate Targets are aligned, each selected Algorithm is used to calculate a score. 1-sequence Algorithms use the match sequence as input. 2-sequence algorithms use the query and the match sequences as input. In addition, new scoring algorithms can be implemented by creating an Algorithm subclass in the source/algorithms subdirectory that utilizes the “universal” Algorithm interface developed for AddTag (sequence, side, target, pam, upstream, downstream).

Nearly all published implementations of CRISPR/Cas scoring algorithms lack flexibility in varying spacer lengths and alternative PAM sequences. We modified the scoring algorithms presented in these papers to allow for assessment of spacers less than or greater than the typical 20 nt length. Additionally, they have been expanded to include scoring of ambiguous characters (using the unweighted average score, a subsampled average, and sometimes the maximum score). For full information on how each Algorithm was adjusted, please refer to the source code in the source/algorithms subdirectory.

### On-target calculations

AddTag evaluates the predicted joint binding and cutting efficiency of a gRNA:RGN complex for the Target sequences it identifies. These are “on-target” scores, and they rely on analyzing the Target site with a model. AddTag implements these as 1-sequence Algorithms. There are 93 known Cas protein families, spread across 394 PSSMs, 2 classes, 6 types, and more than 16 subtypes (Makarova *et al.* 2011; Makarova *et al.* 2015; Shmakov *et al.* 2015; Shmakov *et al.* 2017). AddTag has implemented the field-standard scoring schemes for Cas9 and Cas12a (Cpf1). Several scoring algorithms are provided for use with uncharacterized Cas proteins, such as the “linear” score. However, these are founded on unsophisticated assumptions, such as SPACER to target homology lengths and positions of errors within the alignment relative to the PAM site.

By default, AddTag uses the Azimuth on-target score for use with Cas9 motifs (Doench *et al.* 2016). The Azimuth algorithm takes as input a genomic Target (including the spacer and PAM sequence plus a few nucleotides up- and down-stream of it), and compares it to a gradient-boosted regression trees model trained on cutting efficiency of over 4000 individual sgRNAs targeting sites in 17 genes in human A375 cells. Anecdotal reports from various labs indicate Cas9 cutting efficiency in yeast, mouse, and fly cells mirror these predicted values. For Cas12a motifs, AddTag includes the CINDEL/DeepCpf1 Algorithm, which predicts the likelihood of getting a Cpf1-induced indel at the target locus (Kim *et al.* 2016; Kim *et al.* 2018). Thus, it serves as a decent proxy for an on-target score. CINDEL is a logistic regression classifier trained on 938 Spacer-Target pairs in HEK293T cells. Additional on-target scores implemented are CRISPRater (Labuhn *et al.* 2017), Doench 2014 (Doench *et al.* 2014), Housden (Housden *et al.* 2015a), and Moreno-Mateos (Moreno-Mateos *et al.* 2015).

### Off-target calculations

The off-target score represents the predicted fraction of events that the gRNA:RGN complex will associate with the intended Feature (region of DNA to be targeted) and restrict it compared to all sites with similar homology to the gRNA spacer. Spacers with high off-target scores are preferred, and indicate that unwanted restriction events are unlikely to occur.

Each Target motif is composed of the spacer sequence (SPACER), the restriction sites (), the PAM, and the relative position of the PAM to the spacer sequence (> or <), defined by the following syntax:  
. The Target motif is encoded by the user as a string of characters (Identifying Targets and Feature expansion).

The off-target calculation we use is the general-purpose MIT Guide Score (Massachusetts Institute of Technology 2014) , which can be applied to the results from any number of scoring Algorithms. We calculate it through the following steps:

1. Align the motif to the Feature (or expanded Feature), which is the region to be disrupted, cut, or edited. Each substring in the Feature matching the motif is considered a query   
   .
2. Align each query exhaustively across the gDNA and dDNA expected to be present. Each substring in the genome with homology to the query that passes the postfilter is called a match , and these are put into two categories. Those that lie within the Feature are on-target matches (denoted by ), and those that lie outside the Feature are off-target matches (denoted by ). Thus, each query has a set of within-Feature and outside-Feature matches.

Alignment links each query with a set of matches   
. Furthermore, each match is classified as either within-Feature or outside Feature such that the set of within-Feature matches is  
. And the set of outside-Feature matches is  
.Thus, the mutual exclusivity can be written as  
, and   
.

Each algorithm is contained within the set of all algorithms   
. Each algorithm can take either 1 or 2 sequences as input. The algorithm assigns each match a score such that  
. The final off-target score for any particular query and algorithm pair is calculated as  
. For each off-target compatible algorithm, this procedure is followed to calculate an off-target score.

The final off-target score for any particular algorithm is thus a ratio between the sum of targeting efficiency for acceptable sites and the sum of targeting efficiency across all sites. A higher score means more of the predicted targeting is at the intended Feature. A lower score means more predicted targeting outside of the Feature. The off-target score represents the frequency a gRNA:RGN complex will target a correct site in the genome. AddTag reports final off-target scores as percentages on a scale from 0 to 100 ().

We decided to use the MIT Guide Score due to its wide adoption rather than the alternative Stemmer (Stemmer *et al.* 2015) method for calculating off-target scores. The Stemmer off-target score scales relatively to the number of matches, and thus requires additional computations to compare across experiments. The MIT Guide Score, contrarily, returns a value constrained by probability, so scores can be compared across motifs, Features, and genomes.

Of note is that the off-target specificity is dependent on the number of errors (mismatches, inserts, deletions) permitted by the Aligner used. In the genome editing experiments presented in this paper, we assume that the Aligner finds all relevant matches.

Implemented off-target scores include CFD (Doench *et al.* 2016), Hsu-Zhang (Hsu *et al.* 2013), a simple linear model (this study), CRISPRater (Labuhn *et al.* 2017), Doench 2014 (Doench *et al.* 2014), Housden (Housden *et al.* 2015a), and Moreno-Mateos (Moreno-Mateos *et al.* 2015).

### Spacer ranking

Like CRISPOR (Haeussler *et al.* 2016), AddTag calculates multiple types of Algorithm scores to evaluate how appropriate a Target is for genome editing. This raises the obstacle of effectively ranking Targets in a useful manner that takes all the different scores into account. For our solution, we propose each Algorithm score is given a weight function that transforms the raw score into a weighted one. Then, the final score is the product of all weighted scores (identical to the Primer score weighing implementation in Equation 1 and Equation 2).

By default, AddTag utilizes sigmoidal weight functions that estimate the cumulative density of scores from random Target sequences that match the Target motif. If needed, the user can apply a different weight function or altered weight parameters to each scoring Algorithm. This provides an easily tunable mechanism to specify which scores are most important for any CRISPR/Cas application. For example, the Azimuth (Doench *et al.* 2016) Algorithm returns a score in the inexact domain from 0 to 90, and its corresponding weight function converts the scores to weights on the range from 0 to 1 (Figure S7).

For each candidate Target, the weights of all selected Algorithms are multiplied together, which gives the final, reported product weight (Equation 2). All candidate Targets are then re-arranged in decreasing order from the highest weight to the lowest weight.

Because the sigmoidal weight function approximates the cumulative density of scores, the transformed weight represents the percentile of the input score. Thus, the raw score is converted into a weight that represents how good that score is compared to all other possible scores. This means weights of different algorithms are comparable, and weights of different sequences are comparable as well. Additionally, at the time of publication, each on-target Algorithm implemented displays a unimodal score distribution (Figure S7), which is required for the sigmoidal calculation to approximate the CDF. If the score distribution was multimodal, one additional sigmoidal product should be added for each mode. Thus, the sigmoidal transformation of the Algorithm scores is sufficient for Target ranking.

## Generating knock-out dDNAs that contain AddTags for intermediary genome

The AddTag software generates ΔdDNA sequences with inserts that facilitate several genetic and molecular biology techniques. The general label for these inserts is addtag. AddTag is able to use any arbitrary user-defined sequence as the insert, as well as create several types of specific addtags as follows (Figure S6).

### Mintag

What separates AddTag from other programs that identify RGN binding sites is its ability to create unique gRNA Targets at the site of cleavage. If “mintag” (formerly “mAT” or “mini-add-tag”) is selected as the insert type for the first round of editing, then AddTag generates the Step 1 dDNA by stitching the immediately-adjacent upstream and downstream regions of each +Feature or expanded Feature (eFeature) together into one concatemer. If the junction sequence is not unique in the genome, then it uses a combination of 3 additional adjustments to generate a unique site:

1. Additional bases upstream of the Feature can be trimmed (thereby effectively expanding the Feature)
2. Bases can be added
3. Additional bases downstream of the Feature can be trimmed (also expanding the +Feature)

Each of these can be specified as command line arguments, as well as the final Step 1 dDNA fragment size. Default mintag implementation in AddTag uses “brute force” calculation of all k-mers possible for the insert if the query insert size is less than 5 nt. Otherwise, it performs a random sample of k-mers to obtain sequence suitable to constituting a Target site.

### Addtag

AddTag derives its name from the “addtag” insert type (formerly “AT” or “add-tag”) for the first round of editing. The Step 1 dDNA is a concatemer of the upstream flanking region, an explicit Target sequence that matches the Target motif, and the downstream flanking region. AddTag will procedurally generate this insert so its nucleotide composition differs as much as possible (within a stochastic sampling distribution) from the rest of the genome, thereby minimizing the likelihood the generated Target sequence exists elsewhere, and maximizing the off-target score. This is especially useful when the input genome sequence represents only a portion of the true DNA within the biological system.

### Unitag

The “unitag” insert type is used for generating dDNAs that contain a single Target motif for all edited loci. Because the same insert sequence is added to every dDNA, each Feature edited in the intermediate genome (ΔgDNA) will contain identical unitag sequences. This allows a single gRNA to target every locus in subsequent genome editing steps. Like the addtag product, the unitag is a random sequence that is generated from the complement composition of the genome, thereby increasing the probability of specific RGN activity, even in the absence of a complete genome sequence as input.

## Finding, scoring, and ranking primer designs

### Evaluating single Primers

One central utility of AddTag is to choose appropriate primer sequences that must bind to genomic templates. To do so, AddTag independently computes several attributes of the primer sequence into attribute scores. Then it converts each attribute score into a weight. Finally, it calculates the final primer weight as the product of the attribute weights. AddTag scores the following attributes of the Primer sequence (Figure S9):

1. The %GC of the primer sequence;
2. The length of the primer sequence;
3. The minimum change in Gibbs’ free energy (ΔG) of the primer sequence—although some evidence suggests the effective interference of intended behavior of primers differs based on whether it is a hairpin, homodimer, or heterodimer, we elected to consider only the minimum ΔG reported by the user-selectable software (UNAFold (Markham and Zuker 2008), ViennaRNA (Lorenz *et al.* 2011), Primer3 (Untergasser *et al.* 2012));
4. The melting temperature (Tm) of the primer as a proxy for annealing temperature (Ta), which holds valid only under the assumption the primer lengths are similar and secondary structures non-existent (Rychlik *et al.* 1990; Rychlik 1993; SantaLucia 2007);
5. The maximum 3’ self-complementation length (Rychlik 1993).
6. The 3’ GC clamp length as a computationally-efficient proxy for 3’ end stability (Marky *et al.* 1981; Breslauer *et al.* 1986);
7. The number of G and C residues in the last 5 positions of the primer sequence to serve as a heuristic to minimize off-target hybridization by the 3’ end of the primer;
8. The number of consecutive, repeated nucleotides (also known as run length);

A primer attribute describes an intrinsic property of the primer at experimental conditions (salinities, temperature, and nucleotide concentrations), and is calculated by feeding the primer sequence into a model that produces the attribute score   
. Accordingly, is the set of all primer attribute models  
, and is the set of all attribute scores for a primer  
.

Let be either a canonical or ambiguous nucleotide base  
, and let represent the ordered nucleotide sequence of the primer with length   
.

Each primer attribute score derived from its sequence is passed through a function to scale its quality on a score from 0 to 1, called its weight  
. We define the general formula for weight as a sigmoidal (also called logistic) function to model the desired attributes. Sigmoidal functions are useful because they define thresholds. On one side of the threshold, there is a severe penalty, and on the other side, the penalty is light. Additionally, AddTag implements hard minimum and maximum cutoffs for each attribute score, outside which the weight is set to 0. Attributes 1 and 2 apply bisigmoidal weight functions; attribute 3 applies a unisigmoidal weight function; and attributes 4, 5, 6, 7, and 8 apply uniform weight functions (Figure S9).

Below, we review attribute parameters for a typical bisigmoidal weight function. Each attribute provides a defined set of parameters to transform the score into an attribute weight . First, AddTag defines two thresholds, which we refer to as and   
. Thus, and represent the inflection points of the slopes, and and represent the steepness of the function (Figure S8). For each threshold, we add a sigmoidal factor in the weight calculation

Equation 1

. Please note that any number of sigmoid definitions can be multiplied to produce a complex function for converting score into weight.

Finally, the primer weight for the sequence is calculated as the product of all primer attribute weights

Equation 2

.

For each attribute, represents the broad probability of successful hybridization to genomic DNA using that primer. Higher correspond to higher likelihood of successful binding and thus amplification. Each is treated independently. Thus is a proxy for the joint probability of successful hybridization across all attributes.

### Evaluating Primer Pairs

A number of methods have been proposed to estimate the compatibility of a primer pair (a sum of weighted primer attribute scores in (Kalendar *et al.* 2014) and (Untergasser *et al.* 2012)). We chose to implement the product strategy over the method implemented in the Primer3 software because Primer3 will exclude a large proportion valid primers (Mann *et al.* 2009).

In chemical isolation, several considerations exist for guaranteeing oligonucleotide sequences do not form secondary structures, but can anneal to their template (Figure S9):

* The difference in Tm between forward and reverse primers should be minimal;
* The minimum ΔG of the heterodimer should be as large as possible;
* The amplicon size (2 sigmoidal functions/types) should be within thermodynamically achievable range;
* The maximum 3’ heterodimer complementation length should be as short as possible.

Like individual primers, primer pairs also have a weight that is the product of its attribute weights. The primer pair joint weight is the product of forward primer weight , the reverse primer weight , and the pair-specific attribute weights

Equation 3

. The greater the joint weight, the better the expected amplification. While it has been demonstrated that template sequence composition in between the forward and reverse primers does affect amplification efficiency (McDowell *et al.* 1998; Rose *et al.* 2006; Mamedov *et al.* 2008), we have omitted calculating this for simplicity. For example, if amplification necessitates strand displacement, efficiency can be lowered (Ignatov *et al.* 2014). Furthermore, AddTag does not quantify non-specific transient primer-template binding, which can minimally affect the proportion of time the Primer binds to the intended site. Also no attempt was made to model technical variance in PCR efficiency (Svec *et al.* 2015).

### Evaluating primer sets

Similar to the Primer and Primer Pair data structures, the Primer Set data structure has specific attributes that individually weighted and are multiplied together to form the final weight. The first set of attributes regard the non-redundant list of Primer sequences. We make a non-redundant set of Primers from all PrimerParis. Each PrimerPair has a forward sequence and a reverse sequence.

. We calculate the mean melting temperature , where is the number of elements in

. Next, we calculate as the product of the weight of the difference between and across all elements of

. Next, we calculate one attribute based on the PrimerPair data structure—the product of the joint weights , or if of a PrimerPair is repeated, the average joint weight (as a simplification).

Finally, the weight of a PrimerSet is calculated as follows:

Equation 4

. The weight of the PrimerSet therefore incorporates all PrimerPair joint sequence weights with the addition of an additional melting temperature constraint. This process allows for simpler calculation by relying on pre-computed at the cost of the one replicated attribute involving , thereby artificially increasing its net importance. Because relies on a non-redundant list of sequences, a primer oligonucleotide possessing multiple pairs is only penalized a single time. Also, because relies on a non-redundant list of paired primers, duplicate pairs are effectively counted only once. Together, the weight encourages calculating a minimal set of primers that are compatible with each other.

### Constructing optimal primer sets

Following primer design best-practices (Rychlik 1993; Rozen and Skaletsky 1999; Yuryev 2007; Kalendar *et al.* 2014), we designed the AddTag software to generate PCR primers for confirming the presence or absence of intended editing at the chosen locus. The presence of amplification is usable as both a positive and negative control. Most amplicons are between 400 and 800 nt in length so they can be amplified with minimal changes to PCR conditions. The amplicons resulting from sF/oR and oF/sR amplification span the intended locus on at least one of either the upstream junction or the downstream junction, preferably both. This ensures the Feature/Insert is at the intended location in the genome. Additionally, these pairs enable discrimination of erroneous within-Feature/Insert HDR events. The iF/iR primer pair will amplify a region within the Feature/Insert, thereby indicating its presence somewhere in the genome. The sF/sR Primer pair will amplify only if the Feature is absent at the locus.

AddTag uses a 3-step computational process to find these cPCR primers. (A) AddTag performs *in silico* recombination to generate the expected genomes given successful dDNA integration, and it links subsequent edits to the same locus together into groups. (B) AddTag delimits 4 discrete regions a Primer can be found for each genome in each group, and it identifies all usable Primer Pairs. (C) AddTag performs simulated annealing to identify the best set of primers for each locus group.

1. Before AddTag can find the best primer set, it must first identify regions of DNA shared across all editing rounds that flank the intended locus. The first step to achieving this is by generating the expected genome sequences given successful dDNA integration after each round of genome editing. This process is called *in silico* recombination, and proceeds in a cyclical manner. The following steps are performed for each round of genome editing:
   1. All dDNAs are aligned to the gDNA;
   2. Each dDNA alignment is segregated into “upstream”, “US homology”, “insert”, “DS homology” and “downstream” regions; and
   3. any sequence in the “insert” region of the gDNA is replaced with that of the dDNA.

For simplicity, this process is performed in the order from higher genomic coordinates to lower coordinates, thereby allowing for multiple loci to be edited in a single editing step. After those 3 steps in each round of editing, all dDNA edits that overlap are deemed to be occurring at the same locus, and these are grouped together.

The AddTag program naïvely assumes the expected number of alignments parallels the probability of an HDR event. The E-value describes the expected number of alignments by chance given the scoring scheme, nucleotide composition, alignment length, and genome size (Altschul *et al.* 1990). We therefore assume that a “significant” alignment is one whose E-value is below 1 (E-value < 1). These are the only alignments that are considered for HDR. As the length and percent identity of an alignment increases, the E-value decreases. Most flanking arms will have smaller subsequences that can align to hundreds of places in any given genome, and thus are potential sources of non-target HDR. However, AddTag assumes these hundreds of micro-alignments have negligible impact on the total number of HDR events because these sites are unlikely to have double-stranded breaks. Thus, the E-value of 1 can be thought of as a minimum threshold by which an alignment can be considered significant, and therefore likely to drive HDR.

1. After the locus groups are created, AddTag identifies primer sequences, assesses their usability, and assesses the usability of all required Primer Pairs. It does this in three broad steps.
   1. Four strand-specific regions are identified for each locus: “FAR\_UPSTREAM” where sF Primers reside, “FAR\_DOWNSTREAM” where sR Primers reside, “FEATURE\_F” containing oF and iF Primers, and “FEATURE\_R” containing oR and iR Primers. AddTag uses a sliding window to identify all Primers within each region of each gDNA. Depending on the desired allelic specificity, certain Primers are excluded from the list of potential Primers. For instance, if the Primers need to be multi-allelic, then Primers with identical sequences existing in all gDNA within the region are kept (any Primer whose sequence does not exist in all gDNAs are discarded). The “FAR\_UPSTREAM” and “FAR\_DOWNSTREAM” is processed once for each locus group, but the “FEATURE\_F” and “FEATURE\_R” regions are processed once for each gDNA (the reference genome followed by each predicted edited genome).
   2. Potential Primer sequences are assessed for suitability (Evaluating single primers).
   3. If a primer passes the suitability requirement, then it is considered for pairing. For each locus, a list of potential sF/sR Primer Pairs are assessed for compatibility (Evaluating primer pairs). For each gDNA (the reference genome followed by each predicted edited genome) a list of potential sF/oR, oF/sR, and iF/iR pairs is similarly created.
2. After all lists of potential Primer Pairs are created and assessed for compatibility, those pairs are fed into a simulated annealing procedure to identify the best set of compatible primers, stored in a Primer Set data structure. For a typical two round experiment at a single locus, there is a single amplicon A (sF/sR), followed by the amplicons B, C, and D (sF/oR, oF/sR, and iF/iR, respectively) for the +gDNA, then amplicons B, C, and D for the ∆gDNA, and then amplicons B, C, and D for the AgDNA. This results in a total of 10 Primer Pairs (Figure 4C). For simplicity, AddTag first fixes the sF/sR pair, then cycles through all potential pairs for the remaining Primer Pairs. Note that the AmpF/AmpR Primer Pair determination is separate from the cPCR Primer set determination, and is therefore not included in this section. After using simulated annealing for each sF/sR pair, the results are sorted by first the number of Primer Pairs identified, and second by the weight of the Primer Set.

Briefly, simulated annealing proceeds as follows. The primer design is composed of several lists of Primer Pairs. The goal is to select one element from each list such that all selected elements produce a high weight. The weight of initial selection of elements for each list is iteratively compared with alternative selection of list elements. The process halts once the number of iterations is reached, or a local optimum is determined. AddTag uses a “ranked” simulated annealing process by default, where primer pairs are randomly selected based on their joint weight. A high-speed alternative is included where the lowest-weighted primer pair is always swapped with a higher-weighted pair. Early tests indicate this results in the global optimum at a high frequency (>90% of the time). Each potential selection of primer pairs is stored in a Primer Set object, and the highest-weighted Primer Set is selected for each fixed sF/sR pair (Evaluating primer set). Simulated annealing allows for automatic determination of near-optimal set of compatible primers.

# Supplemental figures

## Figure S1 – 1-step genome editing through CRISPR/Cas-induced HDR requires target disruption

If the gDNA contains the Target, it will be cut, and the endogenous cellular machinery will introduce mutations in an attempt to repair the induced DNA breaks and prevent additional breaks from occurring. Regardless of the dDNA type – ssDNA (A), or dsDNA (B) – the Target will be disrupted in the terminal genome.

1. If input dDNA is ssDNA, and RGN is dsDNA-specific, then the modified Feature can be integrated, but the Target is still disrupted through uncontrolled mutation. Otherwise, the Target is disrupted, and the Feature is not modified as intended.
2. If Target sequence is on the dDNA, then it will be cleaved by the gRNA:RGN complex, thereby preventing dDNA incorporation into the gDNA. Additionally, the Target on the gDNA is disrupted, then repaired erroneously.

Thick horizontal lines represent DNA, with genomic DNA (gDNA) terminating in helices, and donor DNA (dDNA) terminating in blunt ends. Rectangles with internal labels represent annotated regions. Rectangles with staggered edges represent DNA breaks within the annotated region. Annotations with striped shading and labels preceded by an asterisk (✲) represent modified sequences. The stopwatch (⏱) indicates a transient sequence. The Spacer regions of the guide RNAs (Spacer and Scaffold) are color-matched with genomic regions the RNA-guided nuclease (RGN) is programmed to cleave. Black arrows represent gRNA and RGN complex association and restriction of gDNA, followed by either homology-directed repair (HDR) as the process by which dDNA is incorporated into the gDNA, or double-stranded break repair through non HDR methods (NHEJ) (Figure S2) representing uncontrolled mutations to maintain cell viability. The sequence intervening between the Feature selected for editing and the restriction Target is colored gray.

## Figure S2 – Common ways CRISPR/Cas-induced chromosome breaks are repaired

Several classes of mechanisms have been described to repair CRISPR/Cas-induced double-stranded breaks in chromosomes. Panels (A), (B), (C), and (D) depict the first step of genome editing using the AddTag method, where the +Feature and +Target are replaced with the addtag, but each depicts the DNA repaired through a different process. Panel (E) summarizes the expected cPCR amplification of the resulting ∆gDNA.

Thick horizontal lines represent DNA, with genomic DNA (gDNA) terminating in helices, and donor DNA (dDNA) terminating in blunt ends. Rectangles with internal labels represent annotated regions. Rectangles with staggered edges represent DNA breaks within the annotated region. Regions with effectively identical sequences are shaded the same color. Annotations with striped shading and labels preceded by an asterisk (✲) represent modified sequences. The superscript X (X) indicates there are unwanted genomic changes in the terminal gDNA. The stopwatch (⏱) indicates a transient sequence. The Spacer regions of the guide RNAs (Spacer and Scaffold) are color-matched with genomic regions the RNA-guided nuclease (RGN) is programmed to cleave. Black arrows represent gRNA and RGN complex association and restriction of gDNA, followed by a DNA repair process to incorporate dDNA into the gDNA. The sequence intervening between the Feature selected for editing and the restriction Target is colored gray.

## Figure S3 – cPCR amplification of *ADE2CDS* genotypes

*Candida* *albicans* *ADE2CDS* +/+ was replaced with a mintag(CC) to create the ∆/∆ genotype. The mintag(CC) introduced the PAM site for the intermediary AddTag Target, which enabled editing a second time to insert the +gDNA-amplified *ADE2CDS* AFeature at the native locus. For this locus, AddTag returned an optimal primer design lacking ∆iF, ∆iR, ∆oF, or ∆oR primers. Genomic template from colony lysates were added directly to the PCR mix. PCR was conducted for 30 amplification cycles. Hyphens (-) indicate no expected amplification. The sF/sR primer pairs did not amplify in the +/+ and AB/AB genotypes because the extension step was not a sufficient time duration.

## Figure S4 – Schematic of “All-in-1” gRNA stitching methodology

Arrows indicate PCR primers. The AHO1096/AHO1098 primer pair amplifies the universal “A fragment”, and the AHO1099/AHO1097 primer pair amplifies the universal “B fragment”. The AHO1237/AHO1238 primer pair is used in conjunction with a “target-specific” gRNA oligo that bridges between the A and B fragments.

## Figure S5 – Precision editing of transcription factor binding sites enabled by the AddTag method (additional loci)

The AddTag method was used to edit short genomic features, including those that lacked overlapping RGN targets. Segments of Sanger sequencing chromatogram traces are depicted for the experimental target and feature at the edited locus, however the entire region encompassing the step-2 addback dDNA, including the integrative flanks, was verified by Sanger sequencing for each modified strain depicted. Grey bars in the traces represent the Phred quality score from 0 (low) to 62 (high). For step one (purple), the wild-type (+/+) genome was turned into the intermediary (Δ/Δ). For step two (green), the intermediary genome is turned into an add-back (AB/AB) genome.

1. A 9 bp Wor1 binding site (Wor1 bs) that is located upstream of the *WOR1* coding sequence and lacks an overlapping RGN target site was edited via the AddTag method. In step one, both the Wor1 bs and an RGN target 81 bp upstream, along with intervening and flanking sequences included in the expanded feature, were replaced with an AddTag target to create the intermediate *wor1USd* Δ/Δ genotype. Two parallel step two transformations converted the intermediary genome into either an add-back genome (AB0/AB0) containing the wild-type Wor1 bs, or an add-back genome (AB1/AB1) containing an edited Wor1 bs. All sequences outside of the Wor1 bs that were deleted in step one were subsequently restored to their wild-type state in step two.
2. A 14 bp Wor1 binding site (Wor1 bs) that is located upstream of the *WOR1* coding sequence and lacks an overlapping RGN target site was edited via the AddTag method. In step one, both the Wor1 bs and an RGN target 33 bp downstream, along with intervening and flanking sequences included in the expanded feature, were replaced with an AddTag target to create the intermediate *wor1USp* Δ/Δ genotype. Two parallel step two transformations converted the intermediary genome into either an add-back genome (AB0/AB0) containing the wild-type Wor1 bs, or an add-back genome (AB1/AB1) containing an edited Wor1 bs. All sequences outside of the Wor1 bs that were deleted in step one were subsequently restored to their wild-type state in step two.

Thick horizontal lines represent DNA, with genomic DNA (gDNA) terminating in helices. Rectangles with internal labels represent annotated regions. Annotations with striped shading and labels preceded by an asterisk (✲) represent modified sequences. Homologous regions on different DNA molecules are connected by stretched rectangles and represent intended recombination events. The sequence intervening between the feature selected for editing and the restriction target is colored gray. Half-arrows pointing right represent annealing of “forward” primers, and half-arrows pointing left denote annealing of “reverse” primers. Genomic sites where primers anneal are color-matched to their respective primers.

## Figure S6 – Types of AddTag insert sequences generated for Step 1 dDNAs

Thick horizontal lines represent DNA, with genomic DNA (gDNA) terminating in helices, and donor DNA (dDNA) terminating in blunt ends. Rectangles with internal labels represent annotated regions. Each locus is given a different color. The homology regions flanking each +Feature are color-matched with the Feature and addtag. RGN-recruiting AddTag Target sites are labeled with dashed lines.

1. The full-length AddTag and mintag inserts are unique to each locus.
2. The unitag insert is shared among loci.

## Figure S7 – *A priori* weight functions for on-target algorithm scores

By default, AddTag encodes weight functions for on-target algorithm scores that approximate their cumulative density functions. The graphs show the Azimuth (Doench *et al.* 2016) and CINDEL/DeepCpf1 (Kim *et al.* 2016; Kim *et al.* 2018) scores for 100,000 random sequences, used for evaluating Cas9 and Cas12a respectively. The histogram frequency on right vertical axis also corresponds with the weight value (black dashed line). Users can adjust the weight functions for each Algorithm with command line parameters.

## Figure S8 – Description of sigmoidal function parameters

Each sigmoid function is defined by an inflection point and its slope. The standard bisigmoidal function contains a first sigmoid term with the inflection point at and a positive steepness of ; and a second sigmoid term with the inflection point at and a negative steepness of . AddTag sets the final parameter for simplicity. The value represents the Algorithm score or Primer attribute score, and the value is the corresponding weight for that score.

## Figure S9 – AddTag calculates attribute-specific weights from primer and primer pair scores

Primer and Primer Pair attribute scores are converted to attribute weights using either uniform, unisigmoidal, or bisigmoidal functions. Minimum and maximum score cutoffs are indicated by vertical dashed lines. Blue areas under the curve represent bounds of the score domain that yield positive weights. Above each graph is a text example illustrating the attribute, where check marks (✓) indicate a positive weight assignment, and cross marks (🗴), indicate a zero weight; blue text indicates important subsequences for attribute-specific scores.

## Figure S10 – Homologous recombination between gDNA and dDNA

Using traditional 1-step CRISPR/Cas-induced HDR to directly edit a Feature may fail because of microhomology between subsequences within the Feature and a dDNA flanking homology arm. The duplicate US regions thereby decrease genome editing efficiency.

1. In this example, the dDNA contains a subsequence internally that is similar to one of its homology arms (US). Either US region in the dDNA can potentially be selected as a focus for HDR, producing either the intended modification (✲gDNA), or an unintended modification (✲gDNAX). The longer the dDNA sequence, the more likely its homology arms share similarity with subsequences inside the homolog arms. Therefore, longer dDNA sequences are less likely to produce correct edits.
2. Here, the US flanking homology arm exists multiple times on the genomic DNA (gDNA). Each US instance is a potential focus for HDR, producing either the intended modification (✲gDNA), or an unintended modification (✲gDNAX). The longer the expanded Feature, the more likely repeats are shared between dDNA homology arms and eFeature subsequences. Therefore, longer eFeature sequences are less likely to produce correct edits.
3. An example of combinatoric editing under the 1-step method. Here, two Features (Feature 1 and Feature 2) should both be edited. Because they are separated by an intervening sequence (grey) with significant homology to the interior of the dDNA, HDR may not replace Feature 1 with ✲Feature 1 in the modified genome.

Thick horizontal lines represent DNA, with genomic DNA (gDNA) terminating in helices, and donor DNA (dDNA) terminating in blunt ends. Rectangles with internal labels represent annotated regions. Rectangles with staggered edges represent DNA breaks within the annotated region. Regions with effectively identical sequences are shaded the same color. Annotations with striped shading and labels preceded by an asterisk (✲) represent modified sequences. The superscript X (X) indicates there are unwanted genomic changes in the terminal gDNA. The stopwatch (⏱) indicates a transient sequence. The Spacer regions of the guide RNAs (Spacer and Scaffold) are color-matched with genomic regions the RNA-guided nuclease (RGN) is programmed to cleave. Black arrows represent gRNA and RGN complex association and restriction of gDNA, followed by either homology-directed repair (HDR) to incorporate dDNA into the gDNA. The sequence intervening between the Feature selected for editing and the restriction Target is colored gray.

## Figure S11 – Phenotypic characterization of *ZRT2*US strains

Cells from each genotype were cultured under zinc-sufficient and zinc-deficient conditions. 2 independent biological derivations for *ZRT2US* genotypes were plated twice, and representative plate images at 48 hours were selected for depiction. Each row is a different genotype. Each spot originated from 5 μL of culture. Spots on the left-most column of each condition came from cultures with OD600 of 0.3, which were serially diluted by a factor of 10-1 in each successive column to the right. Spots were digitally aligned to a grid undistorted, maintaining original sizes and average spacings.

## Figure S12 – Gene complementation at native and auxiliary loci

Example of loss of function gene complementation at native and auxiliary loci. Each additional locus added for consideration confers a combinatorial expansion for the number of interactions. Therefore, using the native locus eliminates the potential for interference by cis-regulatory elements (CREs), and reduces the potential for interference by unintended mutations. (A) and (B) depict examples where the phenotype is determined wholly by expression of the gene, denoted as the CDS. (C) and (D) depict examples where the phenotype is determined by expression of some unknown gene at the other locus. (A) and (C) show potential results from add-back at the native locus, while (B) and (D) show potential results from add-back at an auxiliary locus.

## Figure S13 – Detailed statistical analysis of the biofilm assay

This figure describes the confidence intervals of the significance tests shown in the main manuscript (Figure 3). Each row, consisting of a pair of horizontal bars in the left panel and an interval in the right panel, represents a Student t-test with unequal variance (Welch). P-values are rounded to the nearest thousandth. The biofilm OD600 between strains are compared according to the alternative hypotheses listed on the left.

## Figure S14 – AddTag ranks targets by combining multiple algorithm scores into a single weight calculation

1. In this example, the +Feature does not contain any valid +Targets, so it is expanded to become an eFeature. The eFeature is scanned for matches to the user-provided Target motif (+Target1, +Target2, and +Target3). Each +Target is scored with each user-selected Algorithm (s1, s2, …), then the scores are converted to weights (w1, w2, …). Note that potential Step 1 dDNA sequences are included in off-target calculations. Finally, +Targets are ranked according to their weight W, which is a combination of the individual Algorithm weights (w1, w2, …).
2. The upstream (US) and downstream (DS) sequences flanking each eFeature, which contains both the +Feature and the +Target, can produce multiple Step 1 dDNAs—each with a different addtag sequence. AddTag scans these potential Step 1 dDNA for AddTag Target sequences. Step 1 dDNAs lacking an AddTag Target are removed from further consideration. Step 1 dDNA that contain valid AddTag Targets are scored and weighed in a manner identical to +Targets, except that Step 2 dDNA is included in off-target calculations.

Thick horizontal lines represent DNA, with genomic DNA (gDNA) terminating in helices, and donor DNA (dDNA) terminating in blunt ends. Rectangles with internal labels represent annotated regions. Black arrows represent computational progression. The sequence intervening between the Feature selected for editing and the restriction Target is colored gray. The progression of the green rectangles represents the annotated regions being scanned for the user-provided Target motif(s).

## Figure S15 – Polymorphism-aware feature expansion

Polymorphism-aware identification of Targets and dDNA flanking homology regions is a five-step process. In this example, three homologous instances of a Feature, designated (A), (B), and (C), are shown in a vertical alignment. Thick horizontal lines terminating in helices represent genomic DNA (gDNA). Rectangles with internal labels represent annotated regions. Targets of the same color have equivalent sequence identity. Different colored Targets have divergent sequences.

1. All Targets matching the user-provided Target motif are identified across each homologous Feature within the maximum bounds of an expanded Feature, designated by the vertical dashed lines. Within the user-specified Feature, there is one multi-allelic set of Targets (Target A2, Target B2, and Target C2). Within the full bounds of the potentially expanded Feature, each allele has four Targets.
2. Based on sequence identity, Targets are grouped into either multi-allelic or allele-specific equivalence groups. In this example, there are five multi-allelic equivalence groups. Because the red Target has one copy in gDNA (A), but two copies each in gDNAs (B) and (C), there are four Target equivalence groups containing red Targets. This is a number equal to the permutations with replacement. A Target whose color is shared between at least two gDNAs precludes it from being allele specific. Thus, there are two allele-specific equivalence groups.
3. For each Target equivalence group, if necessary, the Feature is expanded according to the user-selected expansion method. AddTag provides 5 expansion methods, with “center\_both” as the default (bold). The black vertical line above each annotation represents the origin of expansion. The grey vertical line represents the minimum edge of the eFeature. The horizontal arrows are dashed where they cover the minimum eFeature size, and are solid where they represent further potential expansion. The methods “center\_feature”, “center\_target”, and “center\_both” expand the eFeature in both upstream and downstream directions. The “justify\_feature” and “justify\_target” methods only expand in one direction.
4. The solid red, green, and blue squares on the gDNAs represent a polymorphism at the same genomic position, such as an alignment mismatch. The white squares on the gDNA represent polymorphisms at different genomic positions, such as an alignment gap. The grey shaded regions on either side of the eFeature represent the flanking homology arms that will be present in the Step 1 dDNAs. If the user desires allele-specific Step 1 dDNAs, then AddTag will expand the Feature such that the flanking homology regions contain an adequate number of polymorphisms. In this example, the right side contains an allele-specific gap on each chromosome (3 ≥ 2 polymorphisms minimum), so no expansion is necessary. Without expansion, the flanking homology arm on the left side would contain only 1 polymorphic site. Therefore, the Feature is expanded to the left to encompass 2 sites that are polymorphic across all 3 chromosomes. If the user desires multi-allelic Step 1 dDNAs, then each flanking homology arm should have fewer polymorphisms than the maximum specified (0 polymorphisms maximum). In this example, both the left and right sides of the Feature are expanded so there are no polymorphisms in either flanking homology arm.
5. Because expanded Feature sizes can vary among homologs, the final step is to filter potential sets of expanded Features based on their lengths. In this example, the homologous expanded Features with red multi-allelic Targets all fall within the accepted size range, so it is included in downstream analyses. The putative set of expanded Features with orange Targets has at least one expanded Feature that violates the size thresholds, so this set will be excluded from downstream analyses.

## Figure S16 – Polymorphism-aware primer and primer pair identification

AddTag includes the ability to discriminate between allele-specific, multi-allelic, and allele-agnostic cPCR amplification designs.

1. First, AddTag uses a simple sliding window method to identify all potential primers within the genomic regions that should contain the forward (F) and reverse (R) primers. In this example, 1 valid forward primer is identified, and its sequence is identical in both chromosomes (green color). 3 reverse primers are identified—two allele-specific sequences (red and yellow colors), and one multi-allelic sequence (green color).
2. All potential pairings of forward and reverse primers are assigned allelic specificity.
3. Each potential primer pair is weighed. The final selection of primer pairs for cPCR design proceeds through simulated annealing (Figure 4D).

# Supplemental references

CRISPR Design Tool, pp. Horizon Discovery, Ltd.

CRISPR Specificity Analysis, pp. Horizon Discovery, Ltd.

2019 Benchling, pp.

Aach, J., P. Mali and G. M. Church, 2014 CasFinder: Flexible algorithm for identifying specific Cas9 targets in genomes. bioRxiv.

Altschul, S. F., W. Gish, W. Miller, E. W. Myers and D. J. Lipman, 1990 Basic local alignment search tool. Journal of Molecular Biology 215**:** 403-410.

Bae, S., J. Park and J.-S. Kim, 2014 Cas-OFFinder: a fast and versatile algorithm that searches for potential off-target sites of Cas9 RNA-guided endonucleases. Bioinformatics 30**:** 1473-1475.

Barrangou, R., and Luciano A. Marraffini, 2014 CRISPR-Cas Systems: Prokaryotes Upgrade to Adaptive Immunity. Molecular Cell 54**:** 234-244.

Braglia, P., R. Percudani and G. Dieci, 2005 Sequence Context Effects on Oligo(dT) Termination Signal Recognition by *Saccharomyces cerevisiae* RNA Polymerase III. Journal of Biological Chemistry 280**:** 19551-19562.

Breslauer, K. J., R. Frank, H. Blöcker and L. A. Marky, 1986 Predicting DNA duplex stability from the base sequence. Proceedings of the National Academy of Sciences 83**:** 3746-3750.

Broad Institute, 2015 sgRNA Design Tool, pp.

Camacho, C., G. Coulouris, V. Avagyan, N. Ma, J. Papadopoulos *et al.*, 2009 BLAST+: architecture and applications. BMC Bioinformatics 10**:** 421.

Cornish-Bowden, A., 1985 Nomenclature for incompletely specified bases in nucleic acid sequences: recommendations 1984. Nucleic acids research 13**:** 3021-3030.

Crawford, A. C., L. E. Lehtovirta-Morley, O. Alamir, M. J. Niemiec, B. Alawfi *et al.*, 2018 Biphasic zinc compartmentalisation in a human fungal pathogen. PLOS Pathogens 14**:** e1007013.

Ding, Y., H. Li, L.-L. Chen and K. Xie, 2016 Recent Advances in Genome Editing Using CRISPR/Cas9. Frontiers in Plant Science 7.

Doench, J. G., N. Fusi, M. Sullender, M. Hegde, E. W. Vaimberg *et al.*, 2016 Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9. Nat Biotech 34**:** 184-191.

Doench, J. G., E. Hartenian, D. B. Graham, Z. Tothova, M. Hegde *et al.*, 2014 Rational design of highly active sgRNAs for CRISPR-Cas9-mediated gene inactivation. Nat Biotech 32**:** 1262-1267.

Fu, Y., J. D. Sander, D. Reyon, V. M. Cascio and J. K. Joung, 2014 Improving CRISPR-Cas nuclease specificity using truncated guide RNAs. Nature Biotechnology 32**:** 279-284.

Haeussler, M., K. Schönig, H. Eckert, A. Eschstruth, J. Mianné *et al.*, 2016 Evaluation of off-target and on-target scoring algorithms and integration into the guide RNA selection tool CRISPOR. Genome Biology 17**:** 148.

Hanna, R. E., and J. G. Doench, 2020 Design and analysis of CRISPR–Cas experiments. Nature Biotechnology.

Hodgkins, A., A. Farne, S. Perera, T. Grego, D. J. Parry-Smith *et al.*, 2015 WGE: a CRISPR database for genome engineering. Bioinformatics 31**:** 3078-3080.

Horvath, P., D. A. Romero, A.-C. Coûté-Monvoisin, M. Richards, H. Deveau *et al.*, 2008 Diversity, Activity, and Evolution of CRISPR Loci in Streptococcus thermophilus. Journal of Bacteriology 190**:** 1401-1412.

Hough, S. H., K. Kancleris, L. Brody, N. Humphryes-Kirilov, J. Wolanski *et al.*, 2017 Guide Picker is a comprehensive design tool for visualizing and selecting guides for CRISPR experiments. BMC Bioinformatics 18**:** 167.

Housden, B. E., A. J. Valvezan, C. Kelley, R. Sopko, Y. Hu *et al.*, 2015a Identification of potential drug targets for tuberous sclerosis complex by synthetic screens combining CRISPR-based knockouts with RNAi. Science signaling 8**:** rs9.

Housden, B. E., A. J. Valvezan, C. Kelley, R. Sopko, Y. Hu *et al.*, 2015b Identification of potential drug targets for tuberous sclerosis complex by synthetic screens combining CRISPR-based knockouts with RNAi. Science Signaling 8**:** rs9-rs9.

Hsu, P. D., D. A. Scott, J. A. Weinstein, F. A. Ran, S. Konermann *et al.*, 2013 DNA targeting specificity of RNA-guided Cas9 nucleases. Nat Biotech 31**:** 827-832.

Ignatov, K. B., E. V. Barsova, A. F. Fradkov, K. A. Blagodatskikh, T. V. Kramarova *et al.*, 2014 A strong strand displacement activity of thermostable DNA polymerase markedly improves the results of DNA amplification. BioTechniques 57**:** 81-87.

Kalendar, R., D. Lee and A. H. Schulman, 2014 FastPCR Software for PCR, In Silico PCR, and Oligonucleotide Assembly and Analysis, pp. 271-302 in *DNA Cloning and Assembly Methods*, edited by S. Valla and R. Lale. Humana Press, Totowa, NJ.

Kim, H.-S., and O. Smithies, 1988 Recombinant fragment assay for gene targetting based on the polymerase chain reaction. Nucleic Acids Research 16**:** 8887-8903.

Kim, H. K., S. Min, M. Song, S. Jung, J. W. Choi *et al.*, 2018 Deep learning improves prediction of CRISPR–Cpf1 guide RNA activity. Nature Biotechnology 36**:** 239-241.

Kim, H. K., M. Song, J. Lee, A. V. Menon, S. Jung *et al.*, 2016 In vivo high-throughput profiling of CRISPR–Cpf1 activity. Nature Methods 14**:** 153-159.

Kocak, D. D., E. A. Josephs, V. Bhandarkar, S. S. Adkar, J. B. Kwon *et al.*, 2019 Increasing the specificity of CRISPR systems with engineered RNA secondary structures. Nature Biotechnology 37**:** 657-666.

Kondo, S., and R. Ueda, 2013 Highly Improved Gene Targeting by Germline-Specific Cas9 Expression in *Drosophila*. Genetics 195**:** 715-721.

Labuhn, M., F. F. Adams, M. Ng, S. Knoess, A. Schambach *et al.*, 2017 Refined sgRNA efficacy prediction improves large- and small-scale CRISPR–Cas9 applications. Nucleic Acids Research 46**:** 1375-1385.

Labun, K., T. G. Montague, J. A. Gagnon, S. B. Thyme and E. Valen, 2016 CHOPCHOP v2: a web tool for the next generation of CRISPR genome engineering. Nucleic Acids Research 44**:** W272-W276.

Langmead, B., and S. L. Salzberg, 2012 Fast gapped-read alignment with Bowtie 2. Nature Methods 9**:** 357.

Leenay, R. T., and C. L. Beisel, 2017 Deciphering, Communicating, and Engineering the CRISPR PAM. Journal of Molecular Biology 429**:** 177-191.

Lei, Y., L. Lu, H.-Y. Liu, S. Li, F. Xing *et al.*, 2014 CRISPR-P: A Web Tool for Synthetic Single-Guide RNA Design of CRISPR-System in Plants. Molecular Plant 7**:** 1494-1496.

Li, H., and R. Durbin, 2009 Fast and accurate short read alignment with Burrows–Wheeler transform. Bioinformatics 25**:** 1754-1760.

Lin, Y., T. J. Cradick, M. T. Brown, H. Deshmukh, P. Ranjan *et al.*, 2014 CRISPR/Cas9 systems have off-target activity with insertions or deletions between target DNA and guide RNA sequences. Nucleic Acids Research 42**:** 7473-7485.

Liu, H., Y. Ding, Y. Zhou, W. Jin, K. Xie *et al.*, 2017 CRISPR-P 2.0: An Improved CRISPR-Cas9 Tool for Genome Editing in Plants. Molecular Plant 10**:** 530-532.

Lorenz, R., S. H. Bernhart, C. Höner zu Siederdissen, H. Tafer, C. Flamm *et al.*, 2011 ViennaRNA Package 2.0. Algorithms for Molecular Biology 6**:** 26.

Makarova, K. S., D. H. Haft, R. Barrangou, S. J. J. Brouns, E. Charpentier *et al.*, 2011 Evolution and classification of the CRISPR–Cas systems. Nature Reviews Microbiology 9**:** 467.

Makarova, K. S., Y. I. Wolf, O. S. Alkhnbashi, F. Costa, S. A. Shah *et al.*, 2015 An updated evolutionary classification of CRISPR–Cas systems. Nature Reviews Microbiology 13**:** 722.

Mamedov, T. G., E. Pienaar, S. E. Whitney, J. R. TerMaat, G. Carvill *et al.*, 2008 A fundamental study of the PCR amplification of GC-rich DNA templates. Computational Biology and Chemistry 32**:** 452-457.

Mann, T., R. Humbert, M. Dorschner, J. Stamatoyannopoulos and W. S. Noble, 2009 A thermodynamic approach to PCR primer design. Nucleic acids research 37**:** e95-e95.

Markham, N. R., and M. Zuker, 2008 UNAFold, pp. 3-31 in *Bioinformatics: Structure, Function and Applications*, edited by J. M. Keith. Humana Press, Totowa, NJ.

Marky, L. A., L. Canuel, R. A. Jones and K. J. Breslauer, 1981 Calorimetric and spectroscopic investigation of the helix-to-coil transition of the self-complementary deoxyribonucleotide ATGCAT. Biophysical Chemistry 13**:** 141-149.

Massachusetts Institute of Technology, 2014 Optimized CRISPR Design: Aggregate Scores by Guide, pp.

McDowell, D. G., N. A. Burns and H. C. Parkes, 1998 Localised sequence regions possessing high melting temperatures prevent the amplification of a DNA mimic in competitive PCR. Nucleic Acids Research 26**:** 3340-3347.

Montague, T. G., J. M. Cruz, J. A. Gagnon, G. M. Church and E. Valen, 2014 CHOPCHOP: a CRISPR/Cas9 and TALEN web tool for genome editing. Nucleic Acids Research 42**:** W401-W407.

Moreno-Mateos, M. A., C. E. Vejnar, J.-D. Beaudoin, J. P. Fernandez, E. K. Mis *et al.*, 2015 CRISPRscan: designing highly efficient sgRNAs for CRISPR-Cas9 targeting in vivo. Nature Methods 12**:** 982.

Naito, Y., K. Hino, H. Bono and K. Ui-Tei, 2014 CRISPRdirect: software for designing CRISPR/Cas guide RNA with reduced off-target sites. Bioinformatics 31**:** 1120-1123.

Needleman, S. B., and C. D. Wunsch, 1970 A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology 48**:** 443-453.

Nguyen, N., M. M. F. Quail and A. D. Hernday, 2017 An efficient, rapid, and recyclable system for CRISPR-mediated genome editing in *Candida albicans*. mSphere 2.

Oliveros, J. C., M. Franch, D. Tabas-Madrid, D. San-León, L. Montoliu *et al.*, 2016 Breaking-Cas—interactive design of guide RNAs for CRISPR-Cas experiments for ENSEMBL genomes. Nucleic Acids Research 44**:** W267-W271.

Park, J., S. Bae and J.-S. Kim, 2015 Cas-Designer: a web-based tool for choice of CRISPR-Cas9 target sites. Bioinformatics 31**:** 4014-4016.

Park, J., J.-S. Kim and S. Bae, 2016 Cas-Database: web-based genome-wide guide RNA library design for gene knockout screens using CRISPR-Cas9. Bioinformatics 32**:** 2017-2023.

Perez, A. R., Y. Pritykin, J. A. Vidigal, S. Chhangawala, L. Zamparo *et al.*, 2017 GuideScan software for improved single and paired CRISPR guide RNA design. Nature Biotechnology 35**:** 347.

Raikwar, S. P., N. S. Kikkeri, R. Sakuru, D. Saeed, H. Zahoor *et al.*, 2019 Next Generation Precision Medicine: CRISPR-mediated Genome Editing for the Treatment of Neurodegenerative Disorders. Journal of Neuroimmune Pharmacology 14**:** 608-641.

Rauscher, B., F. Heigwer, M. Breinig, J. Winter and M. Boutros, 2016 GenomeCRISPR - a database for high-throughput CRISPR/Cas9 screens. Nucleic Acids Research 45**:** D679-D686.

Rodríguez-López, M., C. Cotobal, O. Fernández-Sánchez, N. Borbarán Bravo, R. Oktriani *et al.*, 2017 A CRISPR/Cas9-based method and primer design tool for seamless genome editing in fission yeast [version 3; peer review: 2 approved]. Wellcome Open Research 1.

Rose, J. A., K. Komiya, S. Yaegashi and M. Hagiya, 2006 Displacement Whiplash PCR: Optimized Architecture and Experimental Validation, pp. 393-403. Springer Berlin Heidelberg, Berlin, Heidelberg.

Rozen, S., and H. Skaletsky, 1999 Primer3 on the WWW for General Users and for Biologist Programmers, pp. 365-386 in *Bioinformatics Methods and Protocols*, edited by S. Misener and S. A. Krawetz. Humana Press, Totowa, NJ.

Rychlik, W., 1993 Selection of Primers for Polymerase Chain Reaction, pp. 31-40 in *PCR Protocols: Current Methods and Applications*, edited by B. A. White. Humana Press, Totowa, NJ.

Rychlik, W., W. J. Spencer and R. E. Rhoads, 1990 Optimization of the annealing temperature for DNA amplification in vitro. Nucleic Acids Research 18**:** 6409-6412.

SantaLucia, J. J., 2007 Physical Principles and Visual-OMP Software for Optimal PCR Design, pp. 3-33 in *PCR primer design*, edited by A. Yuryev. Springer Science & Business Media.

Shmakov, S., Omar O. Abudayyeh, Kira S. Makarova, Yuri I. Wolf, Jonathan S. Gootenberg *et al.*, 2015 Discovery and Functional Characterization of Diverse Class 2 CRISPR-Cas Systems. Molecular Cell 60**:** 385-397.

Shmakov, S., A. Smargon, D. Scott, D. Cox, N. Pyzocha *et al.*, 2017 Diversity and evolution of class 2 CRISPR–Cas systems. Nature Reviews Microbiology 15**:** 169.

Stemmer, M., T. Thumberger, M. del Sol Keyer, J. Wittbrodt and J. L. Mateo, 2015 CCTop: An Intuitive, Flexible and Reliable CRISPR/Cas9 Target Prediction Tool. PLOS ONE 10**:** e0124633.

Stemmer, M., T. Thumberger, M. del Sol Keyer, J. Wittbrodt and J. L. Mateo, 2017 Correction: CCTop: An Intuitive, Flexible and Reliable CRISPR/Cas9 Target Prediction Tool. PLOS ONE 12**:** e0176619.

Svec, D., A. Tichopad, V. Novosadova, M. W. Pfaffl and M. Kubista, 2015 How good is a PCR efficiency estimate: Recommendations for precise and robust qPCR efficiency assessments. Biomolecular Detection and Quantification 3**:** 9-16.

Swarts, D. C., and M. Jinek, 2018 Cas9 versus Cas12a/Cpf1: Structure–function comparisons and implications for genome editing. Wiley Interdiscip Rev RNA**:** e1481.

Untergasser, A., I. Cutcutache, T. Koressaar, J. Ye, B. C. Faircloth *et al.*, 2012 Primer3—new capabilities and interfaces. Nucleic Acids Research 40**:** e115-e115.

Wilson, L. O. W., S. Hetzel, C. Pockrandt, K. Reinert and D. C. Bauer, 2019 VARSCOT: variant-aware detection and scoring enables sensitive and personalized off-target detection for CRISPR-Cas9. BMC Biotechnology 19**:** 40.

Xu, H., T. Xiao, C.-H. Chen, W. Li, C. Meyer *et al.*, 2015 Sequence determinants of improved CRISPR sgRNA design. Genome Research.

Yuryev, A., 2007 *PCR primer design*. Springer Science & Business Media.

Zetsche, B., Jonathan S. Gootenberg, Omar O. Abudayyeh, Ian M. Slaymaker, Kira S. Makarova *et al.*, 2015 Cpf1 Is a Single RNA-Guided Endonuclease of a Class 2 CRISPR-Cas System. Cell 163**:** 759-771.