Published in final edited form as:

Biotechnol J. 2018 January ; 13(1): . doi:10.1002/biot.201700326.

Designing randomized DNA sequences free of restriction enzyme recognition sites

Audra J. Storm and Paul A. Jensen[†]

Department of Bioengineering and Carl R. Woese Institute for Genomic Biology, University of Illinois at Urbana-Champaign, Urbana, IL, USA

Abstract

DNA libraries containing random “barcodes” complicate synthetic biology workflows that utilize restriction enzymes, since restriction sites can appear inside some barcodes. By removing bases at particular sites in the barcodes, it is possible to create semi-random pools of barcodes that do not contain any restriction sites. The challenge is to remove as few bases as possible to maximize the number of sequences in the pool while ensuring all sequences are free of restriction sites. We present CutFree, a computational approach to create pools of random DNA barcodes that lack a pre-defined set of restriction sites. The resulting pools can be inexpensively produced en masse with standard DNA synthesis techniques. We experimentally validated CutFree by blocking digestion of pools of barcodes designed to frequently contain restriction sites. Using CutFree, we designed a pool of 1.3 billion barcodes that are free from recognition sites for 182 commercially-available restriction enzymes. CutFree is available as a software package and an online tool (<http://jensenlab.net/tools>).

Graphical abstract

Pools of genetic parts containing random DNA barcodes are not compatible with restriction enzyme cloning since the barcodes may contain unwanted restriction sites. The CutFree algorithm

[†]Correspondence to: Paul Jensen, 1206 West Gregory Drive, IGB 3137, MC-195, Urbana, IL 61801, pjensen@illinois.edu.

The authors declare no financial or commercial conflict of interest.

allows users to block restriction sites from DNA pools to prevent unwanted cut sites. The resulting pools can be inexpensively produced en masse with standard DNA synthesis techniques.

Keywords

synthetic biology; bioinformatics; genetic engineering; genomic engineering; next-generation sequencing

Introduction

Restriction enzymes are an indispensable tool for synthetic biology. Cloning methods like Golden Gate assembly use restriction enzymes to efficiently assemble multiple pieces of DNA [1,2]. To enable modular assembly, standardized registries of DNA parts require that sequences are free from a subset of restriction enzyme sites used for assembly [3]. Recently, synthetic biology techniques have been used to generate libraries of genetically engineered organisms. Each mutant in the library contains a unique DNA barcode [4]. After screening, the barcodes are amplified from pools of mutants and sequenced to identify individual mutants [4,5]. Unfortunately, pools of random DNA sequences are not compatible with restriction enzymes, as some of the sequences will contain restriction enzyme recognition sites. These unwanted sites can lead to incorrect assemblies or truncated barcodes. Ideally, semi-random pools of DNA sequences could be designed to contain the largest number of individual sequences but no restriction sites.

Random barcodes are synthesized en masse by adding a mixture of nucleotides during each synthesis step. When designing randomized oligonucleotides (randomers), a set of IUB codes represent the possible combinations of the four DNA nucleotides. The most common codes are A, C, T, and G, which represent the bases individually, and N, which represents all four bases ($N=\{A,C,T,G\}$). There are 15 IUB codes, representing all possible combinations, e.g. $R=\{A,G\}$ and $V=\{A,C,G\}$ (Figure 1A). By judicious choice of codes, a randomized oligo can be restricted such that a particular restriction site never appears in any of the individual oligos. Consider, for example, creating a 20 bp randomer that is free of cut sites for the Type II restriction enzyme BsaI, commonly used in Golden Gate cloning [1,2]. BsaI has recognition site GGTCTC, so replacing every other N in the original randomer with an $H=\{A,C,T\}$ would prevent any sequence from containing two adjacent G's in the forward strand, and therefore, a BsaI site. It is also necessary to block BsaI sites on the reverse strand, which can be similarly accomplished by removing every other C from the forward strand. The resulting BsaI-free randomer, $(DH)_{10}$, contains 3.5×10^9 sequences, only 0.32% of the original 1.1×10^{12} sequences in the starting N_{20} randomer. There are other BsaI-free randomers that allow more sequences, such as $NHHNNHNNNDNNDDNDDNDDN$, which contains 6.2×10^{10} sequences, or 5.6% of the original N_{20} randomer. BsaI-free randomers like $(DH)_{10}$ may be more intuitive to construct manually, but can be less diverse than the most efficiently-blocked randomer.

Restriction sites appear relatively infrequently in randomized pools. For example, a specific six base recognition sequence would appear in ~0.4% of sequences in a random pool of 20 bp oligonucleotides. However, these spurious cut sites often result in truncated products, and

these shorter fragments could become vastly overrepresented due to PCR bias before sequencing. Similarly, a truncated product could confer a growth advantage in a pooled screen, so cells carrying the incorrect product could overwhelm a selection experiment. A method that removes all unwanted restriction sites while maintaining maximum sequence diversity would eliminate these troublesome cases with no additional effort from the experimenter.

In this paper, we describe a method for identifying the most diverse randomizer that is free of recognition sites for any set of restriction enzymes. Our algorithm, named CutFree, converts the problem of blocking restriction sites into a mixed-integer linear program (MILP) that returns a site-free randomizer of maximum diversity. CutFree can be applied to block a randomizer of any length from containing arbitrary sequences. It can also be applied to quasi-random sequences, for example to block cut sites by changing wobble bases in protein coding sequences. CutFree ensures the final sequences do not contain specific subsequences. This approach differs from existing tools for generating random sequences [6,7,8] in two important ways. First, rather than generating thousands of individual sequences, CutFree outputs a single degenerate sequence that can be used to synthesize a pool of oligos with mixed bases. The mixed-base approach eliminates the need for costly large-scale oligo synthesis. Entire pools with mixed bases can be synthesized for the same cost as a single oligo (e.g. a PCR primer), typically a few USD. Second, CutFree guarantees that unwanted sites are removed from all positions in every oligo in the pool. By contrast, other sequence randomizers [6,7,8] constrain overall properties of the sequences, i.e. GC content, codon bias, or amino acid content, so they cannot guarantee that every sequence will be free of cut sites. Completely removing unwanted restriction sites decreases sequence diversity, but CutFree's optimization approach ensures that the maximum number of sequences remain in the pool.

Materials and Methods

Overview of CutFree

Consider the example of designing a 20 bp randomizer (N_{20}) free of EcoRI recognition sequences (Figure 1B). The EcoRI site is palindromic (GAATTC), so a randomizer free of EcoRI sites on the forward strand is also free of sites on the reverse strand. We begin by preventing an EcoRI site from appearing in the first six bases. We identify a set of "blocking codes", or codes that are missing a base from the EcoRI recognition sequence. For example, an H at position 1 is a blocking code, since $H=\{A,C,T\}$ does not contain G. Similarly, B's at positions 2 or 3 are also a blocking codes, since $B=\{C,G,T\}$ and EcoRI requires an A at both positions 2 and 3. For each base there are seven IUB codes that do not contain that base, so there are 42 possible blocking codes to prevent an EcoRI site in the first six bases. We can identify similar blocking codes to prevent an EcoRI site from appearing in bases 2 through 7, and so on through the final potential EcoRI site (bases 15–20).

Having identified the blocking codes for each potential restriction site, we can design a blocked randomizer using combinatorial optimization. We must choose a base for each position (1–20) while requiring at least one blocking code be selected for each potential

restriction site (positions 1–6, 2–7, ..., 15–20). CutFree searches for the optimal codes that satisfy these constraints but maximize the total number of sequences in the randomer.

CutFree formalism

Given a randomer of length m and a restriction site of length k , CutFree creates an MILP with $15m$ binary variables I_{bi} , where b indicates one of the 15 IUB codes (A, C, N, etc.) and $i \in \{1, \dots, m\}$ is the position in the randomer. If $I_{bi} = 1$, the code b appears at position i , i.e. $I_{W4} = 1$ indicates code W has been selected for position 4 in the randomer. We require CutFree to select exactly one code for each location, which we enforce with the m constraints

$$\sum_b I_{bi} = 1, \quad \forall i \in \{1, \dots, m\} \quad (1)$$

We define the set ϕ_b as the blocking codes for base b , so $\phi_G = \{A, C, T, Y, M, W, H\}$ and $\phi_T = \{A, C, G, R, M, S, V\}$. For the EcoRI example, the requirement that at least one blocking code is chosen for the first potential restriction site (positions 1 – 6) is equivalent to the linear constraint

$$\sum_{b \in \phi_G} I_{b1} + \sum_{b \in \phi_A} I_{b2} + \sum_{b \in \phi_A} I_{b3} + \sum_{b \in \phi_T} I_{b4} + \sum_{b \in \phi_T} I_{b5} + \sum_{b \in \phi_C} I_{b6} \geq 1$$

Let r_i represent the i^{th} base in the restriction site. (For EcoRI, $r_1 = G$ and $r_2 = A$.) The above constraint can then be written as

$$\sum_{i=1}^k \sum_{b \in \phi_{r_i}} I_{bi} \geq 1$$

More generally, we can block the restriction site from any of the $m - k + 1$ potential locations with the constraints

$$\sum_{i=1}^k \sum_{b \in \phi_{r_i}} I_{b(i+j-1)} \geq 1, \quad \forall j \in \{1, \dots, m - k + 1\} \quad (2)$$

Any solution to constraints (1) and (2) will not contain a restriction site. Our goal is to maximize the number of possible sequences in the randomer. Let n_b denote the number of bases allowed by code b (for example, $n_A = n_G = n_C = n_T = 1$, $n_W = 3$, and $n_N = 4$). Maximizing the number of sequences is equivalent to maximizing the product of the n_b for the codes selected for each position:

$$\max \prod_{\{I_{bi}=1\}} n_b$$

We can arrive at the same solution by maximizing the logarithm of this objective, which reduces the objective to a linear combination of the binary variables I_{bi} :

$$\max \left(\log \prod_{\{I_{bi}=1\}} n_b \right) = \max \sum_{\{I_{bi}=1\}} \log n_b = \max \sum_{b,i} I_{bi} \log n_b$$

Our MILP approach is easily generalized beyond a single palindromic restriction site. Any number of sites can be blocked by adding a set of constraints (2) for each site. Non-palindromic sites can be blocked from the reverse strand by adding constraints to block the reverse complement from the forward strand. CutFree can also be applied to a semi-random oligo where some of the positions are constrained to particular codes. In this case, constraints are added to force any non-allowed codes to be zero. For example, to require no G's be allowed at position 2, we add the constraint

$$\sum_{b \in \{G,R,K,S,B,V,D\}} I_{b2} = 0$$

MILP problems are often degenerate, with multiple solutions giving the same optimal objective value [9]. Most MILP solvers return only one of the optimal solutions, and the choice from among the optimal solutions is an artifact of the solution path of the solver. When designing randomers, our goal is to find an optimal solution with a diverse set of nucleotides. To increase diversity among the randomers returned by CutFree, we implemented an optional two-step solution. First, CutFree is run as described above to calculate the maximum diversity that can be achieved. Second, CutFree is run again with two small changes. First, a constraint is added requiring the solution to be equal to the optimal objective value. Second, the objective function is changed to a sum of all variables multiplied by random coefficients. This second MILP produces randomers that are still blocked for restriction sites, have the maximum number of allowed sequences, and are randomized. With our two-step method, each run of CutFree produces a new randomer so users can select a randomer that meets other design criteria (balanced GC content, etc.).

It is also possible to have CutFree design randomers that contain multiple blocks per site location. In a randomer with two blocks per site, creating a restriction site requires changing at least two bases. If the right-hand side of the inequality (2) is changed from 1 to an integer $p > 1$, then every potential restriction site will be blocked at least p times. Increasing the number of blocks usually requires a decrease in the number of available sequences, but the resulting randomer is less sensitive to restriction sites created accidentally during oligo synthesis.

Experimental Procedures

All enzymes used in this study were purchased from New England Biolabs, and all oligonucleotides were synthesized by Integrated DNA Technologies. Plasmid DNA extractions were performed with the GeneJET Plasmid Extraction Kit (Thermo Scientific) following the manufacturer's protocol. Chemically competent DH5α cells were prepared

using the Mix & Go Kit from Zymo (T3001) following the manufacturer's protocol. For the transformation, 50 μ L of frozen cells were thawed on ice, and 5 μ L of the plasmid ligation mixture was added to the cells and incubated on ice for 5 minutes. The entire volume of cells was spread across LB plates containing ampicillin (100 μ g/mL) pre-warmed to 37°C.

The pools of barcoded plasmids were created by PCR amplification of the pUC19 plasmid [10] with the forward primers of 5'-(barcode)-GTGCCAGCTGCATTAATGAA-3' and reverse primer, 5'-AGCTGCATGTGTCAGAGG-3'. The barcode random sequences are 5'-HDHDHDHDHDHDHDHDHDHDHDH-3' for the unblocked pool, 5'-HDHRHRHRYRHKMKMDHGHKHRY-3' for the 1-block pool, 5'-HKYRYGWKMKMDCGHDMRMRMRM-3' for the 2-block pool, and 5'-HDHRHRHAATTKMKMDHGHKHRY-3' for the pool with the defined MluCI site. The forward primer contains the semi-randomized barcodes with either the defined MluCI restriction enzyme recognition site, AATT, or blocked for the MluCI site with either zero, one, or two blocks per site. The reverse PCR primer does not contain the semi-randomized barcode region. This ensures that during subsequent rounds of PCR amplification, each individual semi-random barcode would become double stranded (Figure 1C). After amplification, fragments were phosphorylated with T4 PNK, blunt end ligated with T4 DNA ligase, digested with DpnI to destroy the original pUC19, and transformed into chemically competent *E. coli* DH5 α bacteria (Life Technologies). Each colony of transformed bacteria should contain a unique barcoded plasmid. The entire plate was scraped up to create the final pool of barcoded plasmids.

Sequencing of the plasmids was performed by ACGT Inc. using the forward primer, 5'-TTTTGCTCACATGTTCTTTCC-3', and reverse primer, 5'-CATGAGCGGATACATATTTG-3', which flank the barcoded region. The 1.2 kb fragments lacking any MluCI sites outside of the random barcodes (Figure 2B) were created using the primers 5'-TTCGATGTAACCCACTCGTGCA-3' and 5'-TTTGCCGGATCAAGAGCTACCA-3'. Densitometry of the DNA bands (Figure 2C) was performed using ImageJ 1.48 software [11] on four experimental replicates run on the same gel. Densitometry results are reported $\pm$ the standard deviation.

Implementation and Availability

CutFree is implemented as a software package for the R programming language [12]. Our implementation uses the Gurobi MILP solver [13], which is freely available for academic use. Both the source code and a web interface are available at <http://www.jensenlab.net/tools>.

Results and Discussion

To test CutFree, we designed a 23 bp randomer that frequently contained a cut site for the enzyme MluCI (AATT). The randomer (HD)₁₀ contains A and T at every position with alternating G and C. On average, a MluCI cut site appears in 22% of the randomer's 9.4×10^{10} possible sequences. Using CutFree, we designed a blocked randomer, 5'-HDHRHRHRYRHKMKMDHGHKHRY-3'. The blocked randomer contains 2.4×10^8 , or 0.26% of the original sequences, none of which should contain a MluCI site. Because the

starting randomer was designed to maximize the occurrence of MluCI sites, blocking the randomer requires a significant decrease in the randomer's diversity. We created pools of plasmids containing either the original or blocked randomers. Sanger sequencing of the pools confirmed that the randomer region contained a mixture of sequences (Figure 2A). We amplified a 1.2 kb fragment of the plasmid containing the randomer and no other MluCI sites (Figure 2B). Digestion of the unblocked pool with MluCI cut $22.4 \pm 2.9\%$ of the barcodes, consistent with our expectation that 22% of the random barcodes contain a MluCI site. MluCI digestion of the blocked randomers showed no detectable cleavage (Figure 2C). The region of the gel where the cut fragments would appear had intensity equal to $0.68 \pm 0.62\%$ of the plasmids, a fraction not significantly different than the expected 0% ($p > 0.11$, t-test, one sample). The modifications in the blocked oligo have removed the MluCI cut sites. We also used CutFree to design a second randomer with at least two blocks per potential cut site (5'-HKYRYGWKMKMDCGHDMRMRMRM-3') (Figure 2C). The two-block oligo contains 5.3×10^6 sequences (0.01% of the original (HD)₁₀ randomer) and was also free from MluCI cleavage (Figure 2C).

To test the scalability and performance of CutFree, we designed randomers that varied in the randomer length or the number of restriction sites blocked. Restriction sites were selected at random from all commercially-available, palindromic, six-base recognition sequences in REBASE [14]. (Most of the commercially-available enzymes in REBASE have six-base recognition sites, and we observed similar results when blocking sites with four to eight bases.) As expected for an MILP, CutFree's runtime increased exponentially with the number of blocked sites (Figure 3A). Blocking a 20 bp randomer from five sites required, on average, 6.7 ± 6.6 s to identify an optimal solution. Although runtime to the optimal solution depends strongly on the number of sites that must be blocked, we observed little variation in the time to find an optimal solution. For example, blocking a 20 bp randomer from six sites required over 60 s, but a feasible solution was returned within a second. This solution did not contain any unwanted restriction sites and was within 15% of the optimal number of sequences. Like many MILPs, it may not be possible to solve CutFree to optimality for very large problems; however, near-optimal solutions can be found in reasonable time and are often sufficient.

The number of potential sequences in a randomer decreases exponentially as more restriction sites are blocked (Figure 3B). Interestingly, we observed less than a 0.6-fold reduction in number of available sequences for each restriction site added. This reduction is equivalent to changing two bases from N's to a three base code (H, B, V, or D). For each six-base restriction site blocked from the entire 20 bp randomer, CutFree needs to remove as little as a single option from two positions in the randomer, highlighting the efficiency of our optimization-based approach.

CutFree's runtime increases exponentially with the length of the randomer. Blocking a single six-base restriction site requires less than 0.01 s for a 20 bp randomer and 0.1 s for a 100 bp randomer (Figure 3C). The exponential scaling is expected given the underlying MILP formalism. Most DNA barcodes, however, are between 20–30 bp [15], which CutFree is able to process in under 1 s. When designing randomers longer than a few hundred base pairs, we recommend 1.) splitting the randomer into multiple smaller sections, 2.) running

CutFree on each section, and 3.) concatenating the sections. Users will need to check that restriction sites are not created at the junctions between the smaller sections.

As a final test for CutFree, we blocked all palindromic Type II restriction enzymes with 6 bp recognition sites from a 20 bp randomer. CutFree found an optimal solution in less than 20 minutes. The blocked randomer (NDNDGDNDNDGDNDNDGDNN) includes 1.3×10^9 individual barcodes. The 182 enzymes blocked by CutFree include the most commonly used restriction enzymes (EcoRI, BamHI, HindIII, NotI, etc.). DNA constructs incorporating barcodes from this pool are therefore compatible with most commercially-available restriction enzymes. This pool can be used whenever researchers require maximum flexibility in subsequent cloning steps.

Concluding Remarks

Randomized barcoding is a critical part of translating synthetic biology to genome-wide scales [4,5]. CutFree provides an easy method for ensuring large numbers of barcode sequences will be compatible with restriction enzyme assembly methods like Golden Gate cloning [1,2]. CutFree produces the largest pool of sequences without introducing any restriction sites. Typically, the number of sequences produced by a randomer is far greater than the number of barcodes needed for a project. However, the larger the starting pool of sequences, the greater the distance (number of differing bases) between individual barcode sequences [14]. By drawing barcodes from a pool of maximum diversity, the resulting barcoded constructs can be more robustly identified by high-throughput sequencing [14]. Even if only a few hundred barcodes are sequenced, it is beneficial to have the largest pool of possible sequences.

The optimization-based framework behind CutFree is not specific to restriction enzyme sites. Any sequence can be excluded from a randomer using CutFree. The MILP framework discussed here can be used as a starting point for randomizing any biological construct with zero tolerance for the excluded sequence. CutFree does not require the starting sequence to be completely random (all N's), so the algorithm can be used to generate pseudo-random sequences. We believe this feature allows CutFree to be used in a variety of applications, including protein engineering and the design of promoters, terminators, and other DNA regulatory elements.

Acknowledgments

This work was supported by the NIH National Institute of Dental and Craniofacial Research (grant DE026817 to PAJ) and the University of Illinois at Urbana-Champaign.

Abbreviations

IUB	International Union of Biochemistry
MILP	mixed-integer linear program
USD	United States dollars

References

1. Engler C, Kandzia R, Marillonnet S. A one pot, one step, precision cloning method with high throughput capability. PLoS ONE. 2008; doi: 10.1371/journal.pone.0003647
2. Engler C, Gruetzner R, Kandzia R, Marillonnet S. Golden gate shuffling: a one-pot DNA shuffling method based on type IIs restriction enzymes. PLoS ONE. 2009; doi: 10.1371/journal.pone.0005553
3. Canton B, Labno A, Endy D. Refinement and standardization of synthetic biological parts and devices. Nature Biotech. 2008; doi: 10.1038/nbt1413
4. Giaever G, Chu AM, Ni L, Connelly C, Riles L, Véronneau S, Dow S, Lucau-Danila A, Anderson K, André B, Arkin AP, Astromoff A, El-Bakkoury M, Bangham R, Benito R, Brachat S, Campanaro S, Curtiss M, Davis K, Deutschbauer A, Entian KD, Flaherty P, Foury F, Garfinkel DJ, Gerstein M, Gotte D, Güldener U, Hegemann JH, Hempel S, Herman Z, Jaramillo DF, Kelly DE, Kelly SL, Kötter P, LaBonte D, Lamb DC, Lan N, Liang H, Liao H, Liu L, Luo C, Lussier M, Mao R, Menard P, Ooi SL, Revuelta JL, Roberts CJ, Rose M, Ross-Macdonald P, Scherens B, Schimmack G, Shafer B, Shoemaker DD, Sookhai-Mahadeo S, Storms RK, Strathern JN, Valle G, Voet M, Volckaert G, Wang CY, Ward TR, Wilhelmy J, Winzeler EA, Yang Y, Yen G, Youngman E, Yu K, Bussey H, Boeke JD, Snyder M, Philippsen P, Davis RW, Johnston M. Functional Profiling of the *Saccharomyces cerevisiae* genome. Nature. 2002; doi: 10.1038/nature00935
5. Wetmore K, Price M, Waters R, Lamson J, He J, Hoover CA, Blow MJ, Bristow J, Butland G, Arkin AP, Deutschbauer A. Rapid Quantification of Mutant Fitness in Diverse Bacteria by Sequencing Randomly Bar-coded Transposons. mBio. 2015; doi: 10.1128/mbio.0036-15
6. Stothard P. The Sequence Manipulation Suite: JavaScript Programs for Analyzing and Formatting Protein and DNA Sequences. BioTechniques. 2000; 28(6):1102–1104. pmid:10868275. [PubMed: 10868275]
7. Villesen P. FaBox: An online toolbox for FASTA sequences. Molecular Ecology Notes. 2007; 7(6): 965–968.
8. Liu S, Hockenberry A, Lancichinetti A, Jewett M, Amaral LAN. NullSeq: A Tool for Generating Random Coding Sequences with Desired Amino Acid and GC Contents. PLoS Comput Biol. 2016; doi: 10.1371/journal.pcbi.1005184
9. Bertsimas, D., Tsitsiklis, J. Introduction to Linear Optimization. 3. Athena Scientific;
10. Norrander J, Kempe T, Messing J. Construction of improved M13 vectors using oligodeoxynucleotide-directed mutagenesis. Gene. 1983 Dec 26.(1):101–6. DOI: 10.1016/0378-1119(83)90040-9
11. Rasband, W. ImageJ, U. S. National Institutes of Health. Bethesda, Maryland, USA: 1997–2016. <https://imagej.nih.gov/ij/>
12. The R Project for Statistical Computing. <http://www.r-project.org>
13. Gurobi Optimization, Inc. Gurobi Optimizer Reference Manual. 2016
14. Roberts RJ, Vincze T, Posfai J, Macelis D. REBASE – a database for DNA restriction and modification: enzymes, genes, and genomes. Nucleic Acids Res. 2015; 43:D298–D299. [PubMed: 25378308]
15. Faircloth B, Glenn T. Not All Sequence Tags Are Created Equal: Designing and Validating Sequence Identification Tags Robust to Indels. PLoS One. 2012; doi: 10.1371/journal.pone.0042543

Figure 1. Overview of CutFree algorithm and plasmid generation
(A) IUB codes specify combinations of nucleotides. (B) CutFree blocks restriction enzyme recognition at sites 1 and 2 by selecting one code from each blocking set (red or blue). For each base in a potential restriction site, there are seven possible blocking codes that exclude the base. By selecting at least one code for each site (i.e. one blue code and one red code), no EcoRI restriction site can appear in positions 1–7. (C) Plasmids containing random barcodes are generated by PCR. One primer contains a randomized sequenced at the 5' end, producing linear fragments with a randomer at one end. The fragments are phosphorylated and blunt-end ligated to form plasmids for replication in *E. coli*.

Figure 2. Experimental verification of CutFree

(A) Sanger sequencing verifies randomer diversity for unblocked (top) and 1-blocked (middle) barcode pools. Each position in the barcode (bases 9–31, grey box) contains multiple bases, except for position 6 in the 1-blocked pool. This position was constrained to be a G for all sequences by CutFree. Selecting a single transformant produces a non-degenerate sequence corresponding to an individual barcode (bottom). (B) Amplifying one half of the barcoded plasmids produces a 1.2 kb fragment that lacks any MluCI site outside of the random barcode (blue line). Digestion with MluCI will produce two fragments (0.8 and 0.4 kb) if a MluCI site appears in the barcode. (C) Digesting the amplified barcode fragment shows MluCI cleavage only in the unblocked pools. On average, 22% of the barcodes in the unblocked pool should contain a MluCI site. Densitometry on the cleaved fragments shows $22.4 \pm 2.9\%$ of the barcodes in the unblocked pool were cleaved. The absence of MluCI cleavage in the 1- or 2-blocked pools verifies that CutFree removed potential MluCI sites. A separate pool of plasmids were created with a defined MluCI site in the center of each barcode. Digesting these barcodes (“+MluCI site”) confirms the 0.8 kb and 0.4 kb digestion products seen in the unblocked pool.

Figure 3. Performance and scaling of CutFree

The runtime of CutFree increases exponentially as more sites are blocked (A). The number of sequences available in the pool of blocked oligos decreases exponentially as more sites are blocked (B). CutFree's runtime also increases exponentially with the length of the randomer (C). Simulations in (A) and (B) were repeated 20 times for each set of restriction sites (size 1–5). In each simulation, a 20 bp randomer (N_{20}) was blocked from containing 1–5 randomly chosen restriction sites. All restriction sites are palindromic, 6 bp sites drawn from REBASE. Simulations in (C) were repeated 10 times at each randomer length (20–100 bp).