The Dissertation Committee for John Andrew Hawkins Certifies that this is the approved version of the following Dissertation:

Investigations in Integrative and Molecular Bioscience

Committee:

William H. Press, Supervisor

Ilya J. Finkelstein, Co-Supervisor

George Biros

Ron Elber

Oscar Gonzalez

Edward Marcotte

Investigations in Integrative and Molecular Bioscience

by

John Andrew Hawkins

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

May 2018

Acknowledgements

Many thanks to my advisors, William H. Press and Ilya J. Finkelstein, for everything they have done to guide me through my work here. My time in graduate school has been challenging but rewarding, and they are to blame and to thank for both. Thank you to my committee for their time and effort on this dissertation. Thank you to Sara Sawyer and her lab for their help on the Chiroptera project, particularly including Maryska Kaczmarek for her work on the project. Thanks to Jeffrey A. Hussmann for guidance along my path in the Press lab. Thanks to the Finkelstein lab for their help and feedback along the way, in particular to those who collaborated with me on this work directly: Cheulhee Jung, Stephen K. Jones Jr., James Rybarski, Kaylee Dillard, and Fatema A. Saifuddin. Thanks also to our extended collaborators on the CHAMP project: Andrew D. Ellington, Ailong Ke, Cagri A. Savran, and Yibei Xiao.

I would also like to thank the funding sources that have made this research possible, including the CSEM Program and Fellowship for myself, the Peter and Edna O'Donnell Foundation for Dr. Press, and a College of Natural Sciences Catalyst award, CPRIT, the Welch Foundation, the National Science Foundation, and the National Institutes of Health for Dr. Finkelstein.

On the personal side, I want to thank my wife, Christa Walikonis, for her support throughout my time here, as well as my mom, my dad, and my sisters. And of course, I would like to thank the loads of friends who have made graduate school a bearable and even fun experience. There are too many to thank individually, but many thanks to y'all!

Abstract

Investigations in Integrative and Molecular Bioscience

John A. Hawkins, Ph.D.

The University of Texas at Austin, 2018

Supervisor: William H. Press

Co-Supervisor: Ilya J. Finkelstein

Modern biology is going through a revolution of new methods and insights resulting from the new availability of high-throughput DNA sequencing technology. I here present work contributing mathematical and computational methods for gaining insight from large DNA sequencing data sets at three distinct levels.

First, I present a method for improving the accuracy and efficiency of DNA barcodes, short sequences of DNA used to label individual molecules in pooled samples. Many DNA sequencing applications depend on the use of DNA barcodes. However, errors in DNA synthesis and sequencing—substitutions, insertions, and deletions—confound the correct interpretation of these barcodes. I here present Filled/truncated Right End Edit (FREE) barcodes designed for barcode error-correction in the context of a downstream sequence.

Second, I present the Chip-Hybridized Affinity Mapping Platform (CHAMP), a novel technology for repurposing used DNA sequencing chips to study the mechanism and sequence preferences of DNA-binding proteins. Since 2012, the CRISPR family of proteins have gained wide application for their efficiency and ease of use in editing genomes in vivo. Using CHAMP, I, in collaboration with experimentalists in Ilya Finkelstein's lab,

investigated the mechanism and sequence preference of the CRISPR Cascade complex, and discovered a novel periodic lack of sequence specificity in DNA binding. I further determined specific nucleotides important for recruitment of and processing by the nuclease domain, Cas3.

Third, I present a meta-analysis of the order Chiroptera, the order of bats, using the new wealth of DNA sequence information of eighteen bat species. The transcriptome sequencing data for two of these bats—*Hypsignathus monstrosus* and *Rousettus aegyptiacus*, bats associated with studies of the Ebola and Marburg viruses respectively—is novel to this study. Using all this DNA sequence information, I reconstructed a high-confidence Chiropteran phylogeny and found 299 genes with signatures of positive selection, a signature associated with viral antagonism. Further study of these genes may shed light on the mechanism through which several bat viruses relevant to human health hijack the cell, including SARS, Ebola, Hendra, and Nipah.

Table of Contents

List of Tables

List of Figures

Introduction

The world of modern biology has gone through a revolution in the past few years: the revolution of cheap DNA sequencing. In 2000, the first draft of the human genome project was completed, sequencing for the first time nearly all of the human genome. This project took more than 10 years and $2.7 billion to complete[1]. Following this milestone, initial improvements were promising, with progress similar to the exponential benchmark used in computer architecture, Moore's Law[2], which projects roughly a doubling of capacity for the same price every two years (Figure 1). However, between 2008 and 2011, the year I started graduate school, progress in DNA sequencing costs outpaced even Moore's Law. While years of progress had brought the cost of sequencing a genome to $10 million by the end of 2007, by the end of 2011, just four short years later, the cost had plummeted three orders of magnitude to less than $10,000 per genome.

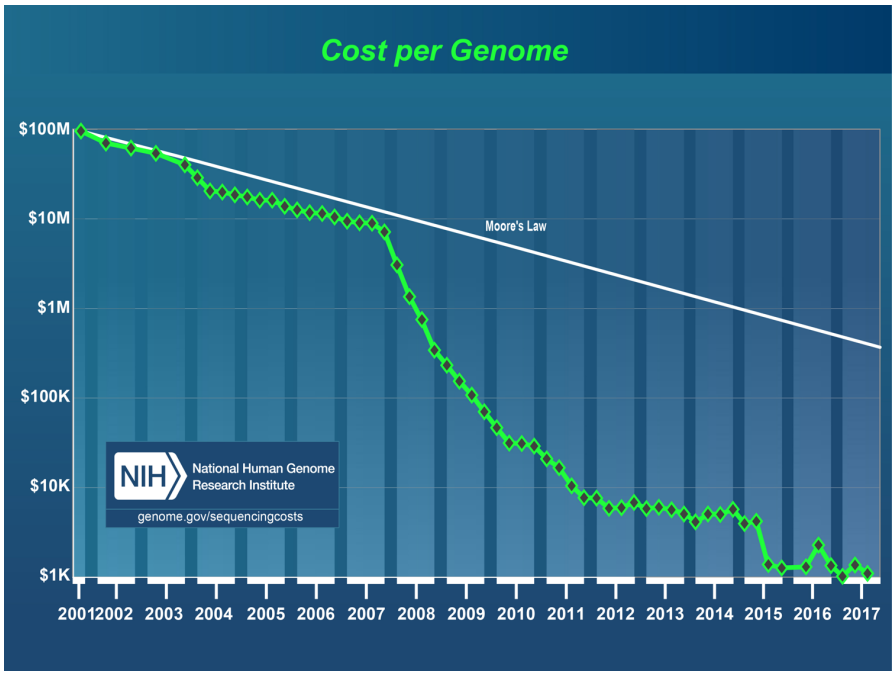


Figure 1: The cost of sequencing a human genome over time.[1]

While this progress is often stated with reference to the cost of sequencing a human genome, the effects of this technology being cheap and broadly available are in fact much more diverse and profound. Cheap DNA sequencing has upended the way questions are asked and the scale of problems that are tackled across all fields of biology, from the study of subcellular structures to the study of whole populations of animals. For example, it has long been known that certain proteins, called transcription factors, initiate the process of transcribing DNA into RNA for use by the cell, but it has historically been very difficult to determine where such proteins bind the DNA on a genome-wide level. Now, with cheap DNA sequencing, technologies such as Chromatin ImmunoPrecipitation and sequencing (ChIP-seq)[3] are able to sequence all locations in the genome which are bound by a transcription factor cheaply and in a single experiment. This has in turn expanded the scale of inquiry to studying not just where it might bind, but where it actually does bind in a wide variety of different specific cell types and conditions[4]. Meanwhile, on the other end of the biology spectrum, DNA sequencing has been used extensively in the tracking of populations of animals. Sequencing hair and droppings samples allows researchers to track whole populations of animals, each animal identified individually, with significantly reduced cost and effort[5]. At all scales of biology, cheap DNA sequencing has empowered new methods for high-throughput data collection.

Turning mountains of new kinds of data into useful insights requires new mathematical models and computational techniques. Depending on overly simple methods can result in overlooking key insights, or in some cases can even lead to exactly the wrong conclusion[6]. Much work has already been done to address the exploding computational needs of the biological community, from sequence alignment[7,8] to whole genome assembly[9–12] to phylogenetic tree inference[13–15]. But much more work remains to be done. For my thesis, I here present work bringing more mathematical and computational methods to bear on DNA sequencing data at three distinct levels: 1) the informatics of DNA sequencing itself, with

the design of short DNA sequences called barcodes with provable error-correction properties, 2) the study of proteins on repurposed DNA sequencing platforms and use of novel data sets to build mathematical models of the functions of gene-editing proteins, and 3) the study of an entire order of species, bats, and their history with viruses as seen through signatures left in their DNA.

Chapter 1: Error-correcting DNA barcodes for high-throughput sequencing*

INTRODUCTION

Many modern large-scale biology experiments use high-throughput DNA sequencing to study the behavior of individual biomolecules in pooled populations. These experiments encode the identity of individual members via DNA barcodes—short, unique DNA sequences that are coupled to each member in the population (Figure 2a). DNA barcode-based identification is central to such diverse applications as single-cell genome and RNA sequencing[16–22], gene synthesis[23,24], high-throughput antibody screens[25,26], and drug discovery[27,28]. Such experiments have been enabled by recent breakthroughs in massively-parallel, pooled DNA synthesis[29,30]. For example, a recent study used DNA barcodes to discover small molecule inhibitors of enzymes by screening ~10^8 small molecules. Each small molecule was attached to a unique set of three DNA barcodes. The highest affinity ligands were enriched via multiple rounds of selection and then identified via high-throughput sequencing of the attached barcodes[31]. The rapid growth of such methodologies in all areas of biomedicine requires the development of large pools (>10^6 members) of unique DNA barcodes to identify individual members (e.g., cells, proteins, drugs) in heterogeneous ensembles.

Every assay with DNA barcodes is subject to errors introduced during DNA synthesis and sequencing. These errors decrease experimental power and accuracy by confounding the identity of individual biomolecules in the population. The most common DNA synthesis

---

* This chapter draws on material from Hawkins JA, Jones SK, Finkelstein IJ, Press WH. Error-correcting DNA barcodes for high-throughput sequencing. (Under review). J.A.H., I.J.F, and W.H.P. designed the research. J.A.H. wrote the software and analyzed the experimental data. J.A.H. and S.K.J. prepared the DNA for sequencing. J.A.H., I.J.F., and W.H.P. wrote the paper. All authors commented on the manuscript.

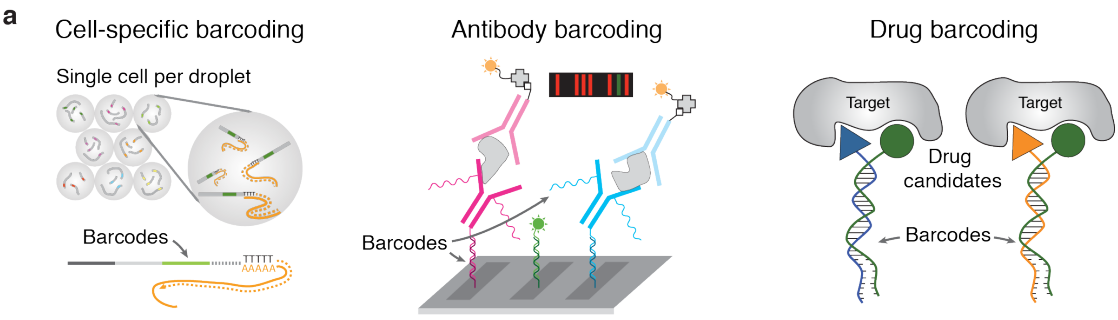
error is a single-base deletion (Results). This is particularly challenging to decode because it causes a frameshift in all downstream sequencing. Substitutions and insertion errors are also common during massively-parallel pooled oligonucleotide synthesis (Results). Our own experimental results are consistent with manufacturer-advertised error rates of up to 1 per 200 nucleotides (nt)[32]. For 20 base pair (bp) long barcodes with no error correction, this translates to a best-case scenario of 10% data lost or, worse, incorrectly interpreted. Next-generation sequencing also has error rates between 10^{-3} and 10^{-4}. This alone represents errors in approximately 1% of our example 20 bp barcodes, which can be limiting for detection of rare events. These errors can be overcome through the use of error-correcting DNA barcodes—DNA sequences that can correctly identify the underlying individuals in a pooled experiment even in the presence of sequencing and synthesis errors.

Error-correcting barcodes must efficiently detect and correct all DNA sequencing and synthesis errors. Many current DNA barcode strategies repurpose error-correcting codes developed for computers[33,34], such as Hamming or Reed-Solomon codes, to DNA applications[35,36]. Hamming distance, i.e., the number of substitutions between two sequences of equal length, is possibly the most used due to its simplicity. However, nearly all well-studied error-correcting codes developed in computer science—including the widely-used Hamming codes—were not designed to handle deletions and insertions, which are the most common errors in DNA synthesis. Such codes are generally used to only detect errors without correcting them, but even then there is a possibility that a single error (e.g., deletion) can convert one barcode into another. Levenshtein codes, also known as edit codes, can theoretically account for all three types of common error: substitutions, insertions, and deletions, but only when the corrupted length of each barcode after errors is known[37,38]. This is a critical limitation in real-world DNA barcode applications because errors can change the barcode length unpredictably, which leads to erroneous decoding of Levenshtein-based barcodes in the context of a longer read (Figure 2b). As a workaround, Levenshtein codes can be used at twice the level of error correction as desired for a given

application, for example using a 2 error-correcting code when a 1-error correcting code is desired, but this is inefficient and significantly decreases the number of valid barcodes for a given oligonucleotide length. In sum, existing DNA barcode strategies are unable to efficiently detect and decode real-world errors encountered during DNA synthesis and sequencing.

Here, we develop and experimentally validate error-correcting Filled/truncated Right End Edit (FREE) barcodes. FREE barcodes can correct substitutions, insertions, and deletions even when the edited length of the barcode is unknown. These barcodes are designed with experimental considerations in mind, including balanced GC content, minimal homopolymer runs, and no self-complementarity of more than two bases to reduce internal hairpin propensity. We generate and include lists of barcodes with different lengths and error-correction levels that may be broadly useful in diverse high-throughput applications. For each barcode set, we calculate hairpin melting temperatures which can be used to select subsets of barcodes to match experimental conditions. Our largest barcode list includes $>10^6$ unique error-correcting barcodes usable in a single experiment. Moreover, appending two or more barcodes together combinatorially increases the total barcode set, producing $>10^9$-10^{12} unique error-correcting DNA barcodes. The included software for creating new barcode libraries and decoding/error-correcting observed barcodes is fast and efficient, decoding >120,000 barcodes per second with a single processor, and is designed to be user friendly for a broad biologist community.

a



b

Error Type	Substitution	Insertion	Deletion
Observed sequence after a single edit	GATTACA → sub. → GCTTACA	GATTACA → ins. → GTATTAC**A**	GATTACA → del. → GTTACA**C**
		Phantom deletion: "A" shifted out of word frame	Phantom insertion: "C" shifted into word frame from rest of read
Hamming distance	1	5	5
Levenshtein distance	1	2	2
FREE divergence	1	1	1

c

	Typical example	Symmetric to Previous	More edits, same free div.	Indels can have zero cost	Changes past end can matter
Expected (E)	CGAT	TAGA	TAGA	GCGC	TAGAT
Actual edits	sub. A GAT ins. trunc. TAGA	del. fill AGAT sub. CGAT	del. fill AGAT del. fill GATC ins. trunc. CGAT	del. fill CGCG del. fill GCGC	ins. trunc. TACGA sub. TACGC del. fill ACGC
Net fill/trunc.	Truncate T	Fill T	Fill T	Fill GC	-
Observed (O)	TAGA	CGAT	CGAT	GCGC	ACGC
FreeDiv(E, O)	2	2	2	0	3
Lev(E, O)	3	3	3	0	3
Hamming(E, O)	4	4	4	0	3

Figure 2: Applications and error-correction strategies of DNA barcodes.

Figure 2: Applications and error-correction strategies of DNA barcodes.

a. Illustrative examples of high-throughput sequencing assays that require large lists of error-correcting DNA barcodes. Barcodes are used to identify individual cells or molecules in pooled libraries (Klein, 2015; Fan, 2008; Melkko, 2004). b. Current strategies to correct synthesis and sequencing errors in DNA barcodes are confounded by insertions and deletions. Hamming distance can only handle substitutions. Levenshtein distance is confounded by the fact that barcodes are prepended to other sequences of interest. Indels thus produce phantom Levenshtein distance errors when bases from the remaining DNA molecule shift into or out of the barcode window. c. Examples of FREE divergence (this work) given the actual edit history. Levenshtein and Hamming distances are also shown for comparison. A substitution and insertion are correctly attributed as 2 edits by FREE divergence (first column). FREE divergence is a symmetric function, i.e., FreeDiv(E, O) = FreeDiv(O, E) (first and second columns). Different actual edit paths can result in the same observed sequence (second and third columns). Indels can have zero cost, particularly near the end of the barcode where they can occasionally be undone by fill or truncation (fourth column). Edits past the barcode end can matter since the fill/truncation step happens only upon observation (fifth column).

RESULTS

Overview of Filled/truncated Right End Edit (FREE) Divergence

After DNA synthesis and sequencing, a barcode of length *n* can be altered, and is not guaranteed to end after exactly *n* bases. Our goal is to design barcodes that can be unambiguously identified from the first *n* bases of the sequenced read. To begin, we define a *filled/truncated right-end m-edit*, hereafter written "FRE m-edit," of a DNA sequence of length *n* to be the result of any *m* edits—substitutions (*sub*), insertions (*ins*), or deletions (*del*)—followed by truncating or filling with any random bases on the right (as from the unknown downstream read) as necessary to return to original length *n* (Figure 2b). For any two DNA sequences *X* and *Y* of the same length, we define the *Filled/truncated Right End Edit (FREE) Divergence* between *X* and *Y,* written *FreeDiv(X, Y),* to be the minimum *m* such that *Y* is a FRE *m*-edit of *X*.

Figure 2c shows a typical example of how FREE divergence captures the actual number of barcode edits in the context of a longer read. An insertion has caused the final T to move out of the barcode window, but FREE divergence correctly accounts for its loss. FREE divergence is a symmetric function, i.e. *FreeDiv(X, Y) = FreeDiv(Y, X)* (Figure 2c). This is because reversing the edits and reversing the right-end fill or truncation step moves one from *Y* back to *X* in the same minimum number of steps, proved below. FREE divergence is defined as the minimum number of steps between the expected and observed barcode, but it is possible to accomplish the same transformation with more edits, for example via the identity *ins-del = sub* (Figure 2c). Also, insertions and/or deletions (indels) near the end of the sequence can result in a FREE divergence of zero if the inserted or filled bases match the truncated or deleted bases respectively. While Figure 2c shows this for deletions, inserting 'GC' instead of deleting it results in the same sequenced barcode. Finally, we note that FREE divergence is not a metric—a mathematically precise term for distance—because edits outside the barcode window can lead to violation of the triangle inequality,

as we show below (Figure 2c). This requires us to use specialized code generation techniques that do not rely on the properties of a metric, and also underlies usage of the term divergence rather than distance throughout this work.

Calculating FREE divergence

FreeDiv(*X,Y*) can be efficiently calculated with a modified Needleman-Wunsch algorithm[39], where the last row and column of the matrix have zero penalty for insertion and deletion corresponding to right-end fill or truncation respectively.

Symmetry and minimum paths

FREE divergence is symmetric because any minimum filled/truncated right end edit path (FREE path) is invertible by inverting all the edits and then inverting the fill/truncation step. Substitutions are invertible with substitutions, while insertions and deletions are invertible with each other in the natural way, so edits by themselves are invertible. Invertibility with the fill/truncation step is less obvious, and requires no edit be truncated off the end. For example, a substitution in the last position followed by any insertion results in the substitution getting truncated off the end. Minimum FREE paths never have any edits truncated off the end, because any truncated edit can be omitted to create a shorter edit path.

Let X and Y be barcodes and let P be any minimum FREE path from X to Y . If P has no fill or truncation, then the fill/truncation step is trivially invertible by doing nothing. Suppose P has a fill step which fills f bases at the end. Then starting at Y and inverting the edits results in exactly those f bases being outside the barcode window, so they are truncated to arrive at X. Suppose P has a truncation step which truncates t bases. Since P is a minimum edit path, none of the truncated bases were edited bases, so they are not needed for the inverted edit path starting at Y . After inverting the edits, t bases need to be

filled, which we fill with the last t bases of X. Hence, any minimum FREE path can be inverted in the same number of edits. Furthermore, since all minimum FREE paths from X to Y and from Y to X are invertible, $FreeDiv(X, Y) \leq FreeDiv(Y, X)$ and $FreeDiv(Y, X) \leq FreeDiv(X, Y)$. Therefore, $FreeDiv(X, Y) = FreeDiv(Y, X)$ and any inverted minimum FREE path is itself a minimum FREE path.

FREE divergence is not a metric

We use the counter-example shown in the right column of Figure 2c. For *FreeDiv*(*TAGA, ACGC*), the modified Needleman-Wunsch algorithm described above produces

$$\begin{array}{c} \\ \mathrm{T} \\ \mathrm{A} \\ \mathrm{G} \\ \mathrm{A} \end{array} \begin{array}{c} \begin{array}{ccccc} \mathrm{A} & \mathrm{C} & \mathrm{G} & \mathrm{C} & \end{array} \\ \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 1 & 2 & 3 & 4 \\ 2 & 1 & 2 & 3 & 4 \\ 3 & 2 & 2 & 2 & 3 \\ 4 & 3 & 3 & 3 & 3 \end{pmatrix} \end{array},$$

so, from the value in the last row and column, *FreeDiv*(*TAGA, ACGC*) = 3. Hence, the following is a minimum filled/truncated right end edit path (FREE path) between TAGA and ACGC:

$$TAGA \xrightarrow{ins.} TACG|A \xrightarrow{sub.} TACG|C \xrightarrow{del.} ACGC$$

The vertical bars ("|") show the end of the barcode window, though the truncation step would not happen until after all actual edits. Now, the above FREE path shows that *FreeDiv*(*TAGA, TACG*) = *FreeDiv*(*TACG, ACGC*) = 1. But *FreeDiv*(*TAGA, ACGC*) = 3, a violation of the triangle inequality.

We note that a previous paper attempted to solve this problem by defining Sequence-Levenshtein codes, but the code generation technique depended on the Sequence-

Levenshtein distance function being a metric, which it is not[40]. The resulting codes often decode erroneously as a result.

FREE Codes

With FREE divergence defined, building an error correcting barcode list is conceptually equivalent to packing spheres in the space of possible barcodes (Figure 3a). We set a barcode length n and call any DNA sequence of length n a word. For any word B, we call the set of all words W such that $FreeDiv(B, W) \leq m$ the *m-error decode sphere* of B, written as $DecodeSphere_m(B)$, or just $DecodeSphere(B)$ if m is clear from context. Any observed DNA sequence within $DecodeSphere(B)$ will by definition decode to (error-correct to) the center word B (Figure 3a.). Then, an m-error correcting FREE code is simply any set of barcodes such that the m-error decode spheres of all barcodes are disjoint, i.e., no two decode spheres overlap. Any corrupted barcode with up to m errors is thus in the decode sphere of exactly one barcode and can be decoded (error-corrected) uniquely (Figure 3a).

Requiring disjoint decode spheres places a limit on the relationship between allowed m, the number of correctible errors, and n, the barcode length: to fit more than one non-overlapping decode sphere in the space requires that $2m + 1 \leq n$. Proof: Suppose the contrary. Let $L \leq 2m$ be the length of the barcode. Then by definition every barcode is at most L substitutions from any other barcode by substituting all of the bases. For any two barcodes B_1 and B_2 define B_{mid} to be the barcode with the first m bases of B_1 and the remaining $L - m \leq m$ bases of B_2. Then $B_{mid} \in DecodeSphere_m(B_1)$ and $B_{mid} \in DecodeSphere_m(B_2)$. Since B_1 and B_2 were arbitrary, it is thus impossible to have two disjoint decode spheres. Therefore, it is impossible to have a non-trivial (i.e. more than one barcode) m-error correcting code of length less than $2m + 1$ bp.

Efficient FREE barcode generation and decoding

A software library accompanying this manuscript efficiently generates FREE barcodes with a given total length and error-correction level. The generation algorithm is conceptually very simple: iterate through the space of n-mers alphabetically, find the decode sphere for each candidate barcode, and reserve barcodes when their decode spheres do not overlap the decode spheres of any previously reserved barcodes (Figure 3a). This set of reserved barcodes by definition forms a valid FREE code. Additional algorithmic details make the process faster and more memory efficient (Methods). Adding valid code words in alphabetical order is a heuristic method previously observed to efficiently pack spheres[41]. Experimental synthesis and sequencing limitations are also incorporated during barcode selection. Candidate barcodes must have: (1) balanced GC content (40-60%); (2) no homopolymer triples (e.g., AAA); (3) no GGC (a known Illumina-based error motif[42]); and (4) no self-complementarity of >2 bases to reduce hairpin propensity. All of our software is available in the GitHub repository accompanying this manuscript (https://github.com/finkelsteinlab).

The number of available error-correcting barcodes for a DNA sequence of length n will depend on the experimentally-required degree of error-correction (Figure 3b). We generated libraries of single-error correcting codes up to a 16-nucleotide length, containing >1,600,000 barcodes. In addition, we generated more robust, double-error correcting codes up to a 17-nucleotide length with >23,000 unique members (Table 2). Barcodes correcting m errors require length at least $2m + 1$ bp, as shown above. Thus, the 1-error and 2-error correcting barcode libraries have minimum lengths of 3 bp and 5 bp respectively. The barcode decoding software runs in time proportional to the length of the barcodes but constant with respect to the number of barcodes in the library. Hence, 1-error and 2-error correcting codes decode at the same speed for a given barcode length even though the 1-error libraries contain many more barcodes (Figure 3c). Even the slowest decodes

considered here, the 17-mer double-error correction barcodes, decode at >120,000 barcodes · sec^{-1} on a desktop computer using a single processor.

Comparison with current error-correcting DNA barcode strategies

Current state-of-the art error correcting DNA barcoding applications often use Hamming or Levenshtein error-correction strategies[35,38]. Hamming codes only correct substitutions, and are thus insufficient for any DNA barcode applications with indels[43]. However, they are linear codes, meaning the code words form a well-structured lattice in barcode space. We tested an alternative hypothesis that pruning these well-packed Hamming decode spheres to subsets with disjoint FreeDiv decode spheres could result in a more efficient packing—more barcodes for a given barcode length—than our alphabetical generation strategy.

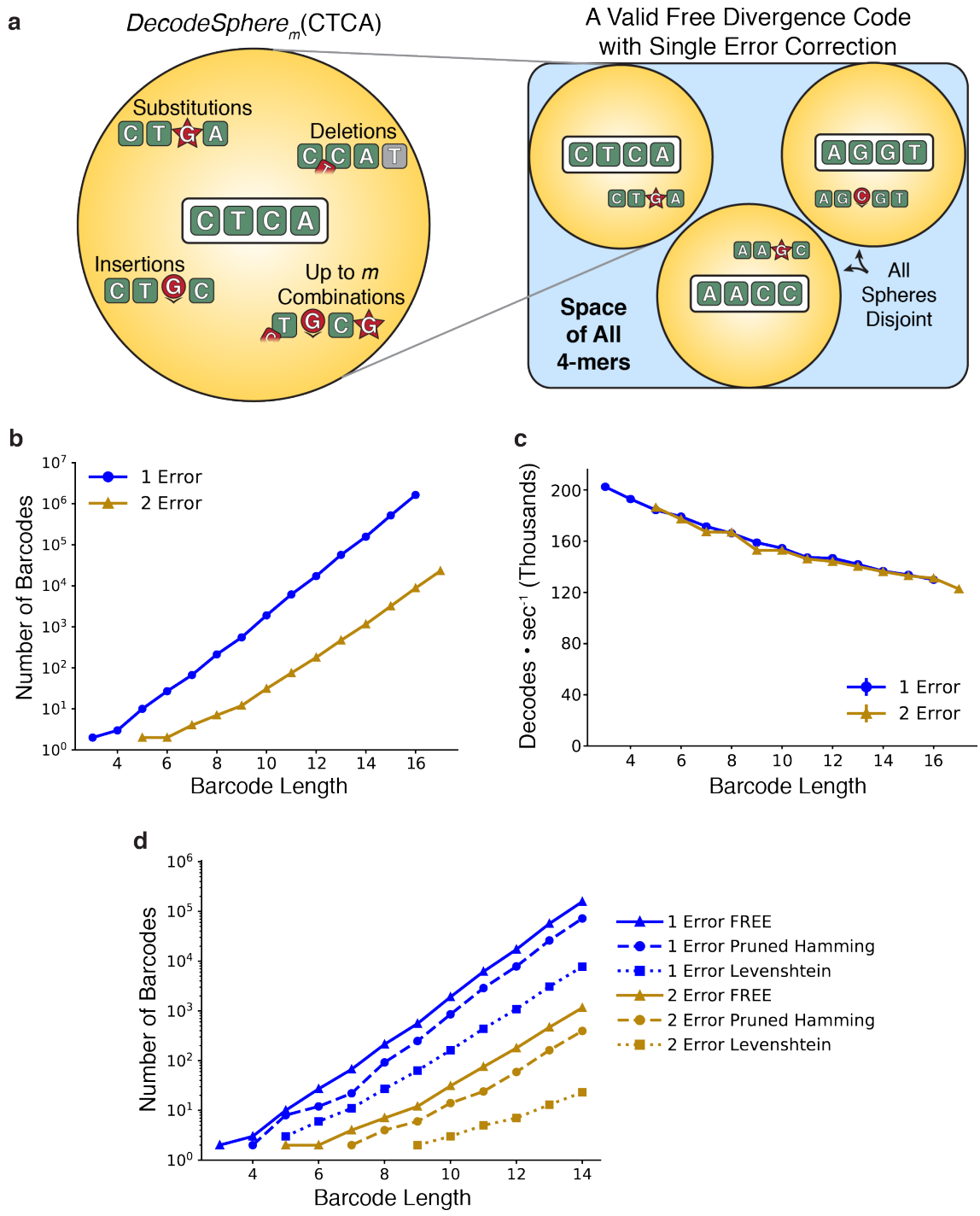


Figure 3: FREE barcode generation and decoding.

Figure 3: FREE barcode generation and decoding.

a. Error-correcting barcode generation is a sphere packing problem. Around each accepted barcode B (e.g., “CTCA”), we reserve DecodeSpherem(B), the set of all sequences within FREE divergence m of B. That is, the set of all sequences with any combination of up to m errors from B, followed by fill or truncation as necessary. Any set of disjoint decode spheres is a valid FREE code (right). b. The number of single- and double-error correction barcodes generated for a range of barcode lengths. c. The accompanying software decodes more than 120,000 barcodes per second for all barcode lengths considered here. d. Comparison of FREE barcode counts against pruned Hamming codes and Levenshtein codes. Hamming codes were pruned to remove members that did not decode FREE divergence errors, while Levenshtein codes were produced at double the error-correction levels for the same purpose. FREE codes produce more barcodes than either of the other methods for all barcode lengths.

Generating a linear Hamming code for DNA strings of length n encoding raw messages of length k and which corrects up to e errors is equivalent to finding a parity check matrix $H = (-P^T | I_{n-k})$ over Galois field $\mathbb{F}_4$ such that any subset of $2e$ columns is linearly independent[44]. Given such a matrix H, all barcodes can be expressed as mG, where m is any raw message vector of length k and G is the generation matrix $G = (I_k | P)$. We found matrices P_1 and P_2, corresponding to single- and double-error correcting linear Hamming codes, via lexicographical search through possible columns of H, accepting new columns if they were linearly independent from all previous subsets of $2e - 1$ columns. We found such P matrices for k up to length 100. The submatrices corresponding to codes up to length 14, as used in Figure 3e, are given by

$$P_1 = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 1 & 3 \\ 1 & 0 & 1 \\ 1 & 0 & 2 \\ 1 & 0 & 3 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 1 & 3 \\ 1 & 2 & 0 \end{pmatrix}, \qquad P_2 = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 2 \\ 0 & 1 & 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 3 \\ 1 & 0 & 1 & 0 & 2 & 3 \\ 1 & 0 & 1 & 2 & 3 & 1 \end{pmatrix}.$$

The resulting linear Hamming codes were then pruned to subsets of valid FREE barcodes. We found that FREE codes generated using our heuristic lexicographic heuristic have about a factor of two more barcodes for a given length than our best pruning of Hamming codes (Figure 3d).

Levenstein codes can be used directly (i.e., without pruning) because they account for indels, but must be used at 2-fold higher error correction for DNA barcode applications

(Figure 2b). We generated such over-corrected Levenshtein barcode sets in a manner similar to the FREE code generation strategy. This strategy produced even fewer barcodes than the pruned Hamming code sets. (Figure 3d, Methods). In sum, FREE codes offer a substantially larger number of usable barcodes for a given barcode length, when taking into consideration real-world errors such as deletions, insertions, and substitutions that are encountered during DNA sequencing and synthesis.

Sphere packing bounds and code efficiency

The optimal packing for an error-correcting code is not known in general. Typical code generating algorithms, including ours, are instead heuristics for finding relatively good codes. For reference, one can usually find a (typically impossible) upper bound on the maximum number of code words by calculating the volume of the space divided by the volume of a single decode sphere. This calculation is complicated here, however, by the fact that FREE divergence decode spheres do not have uniform volume due to degeneracy of insertions and deletions. For example, the sequence AACT only has three unique deletions because a deletion of either A generates the same resulting sequence. Figure 4a shows sphere volumes of 1- and 2-error codes for all words and for only valid code words after our FREE code synthesis and sequencing filters (no homopolymer runs, no triplet complementarity, etc.) for barcodes of length up to 12 bp.

To find the sphere packing upper bound, then, we sorted the volumes of every sphere in the space and found the minimum number of barcodes at which the cumulative sum of barcode sphere volumes is smaller than the space. These upper bounds are shown in Figure 4b, while the corresponding maximum sphere volumes are shown as black bars in Figure 4a. The lower bound for sphere packing of a given code is the best efficiency achieved by any code generation method to date, which for FREE codes is simply the number of

barcodes reported in this paper. The actual maximum possible number of barcodes is somewhere between the two.

Code efficiency is measured, where possible, in terms of a code rate, defined as the number of usable "message" bits that can be encoded in a single barcode divided by the actual number of bits in the sent barcode. In many standard codes, k message bits have r bits added for error correction, giving a code rate of $k/(k + r)$. For n-mer barcodes, each sent base is two bits of information, so the denominator is $2n$. The numerator is the effective number of message bits: the length of the largest binary number smaller than the number of barcodes, given by $\lfloor \log_2(\textit{Number of barcodes}) \rfloor$. However, for our purposes the number of message bits does not need to be an integer, so we will refer to the previous as the actual message bits, while we are more interested in the "raw" message bits: $\log_2(\textit{Number of barcodes})$ without a floor function. These correspond to raw and actual code rates, shown in Figure 4.

The code rate of FREE codes increases with barcode length, and appears to asymptotically approach a maximal code rate determined by the properties of the decode sphere packing. We observe in Figure 3b that after some boundary effects at short barcode lengths, the number of raw message bits
(log of the number of barcodes) increases linearly with the length of the barcodes. The slope of this line, up to a factor of 2 for the x-axis due to using base-4 instead of base-2, is an empirical estimate for the asymptotic code rate—message bits over sent bits—for our packing method. We show estimated asymptotic values for our single- and double-error correcting codes as dashed lines in Figure 4c.

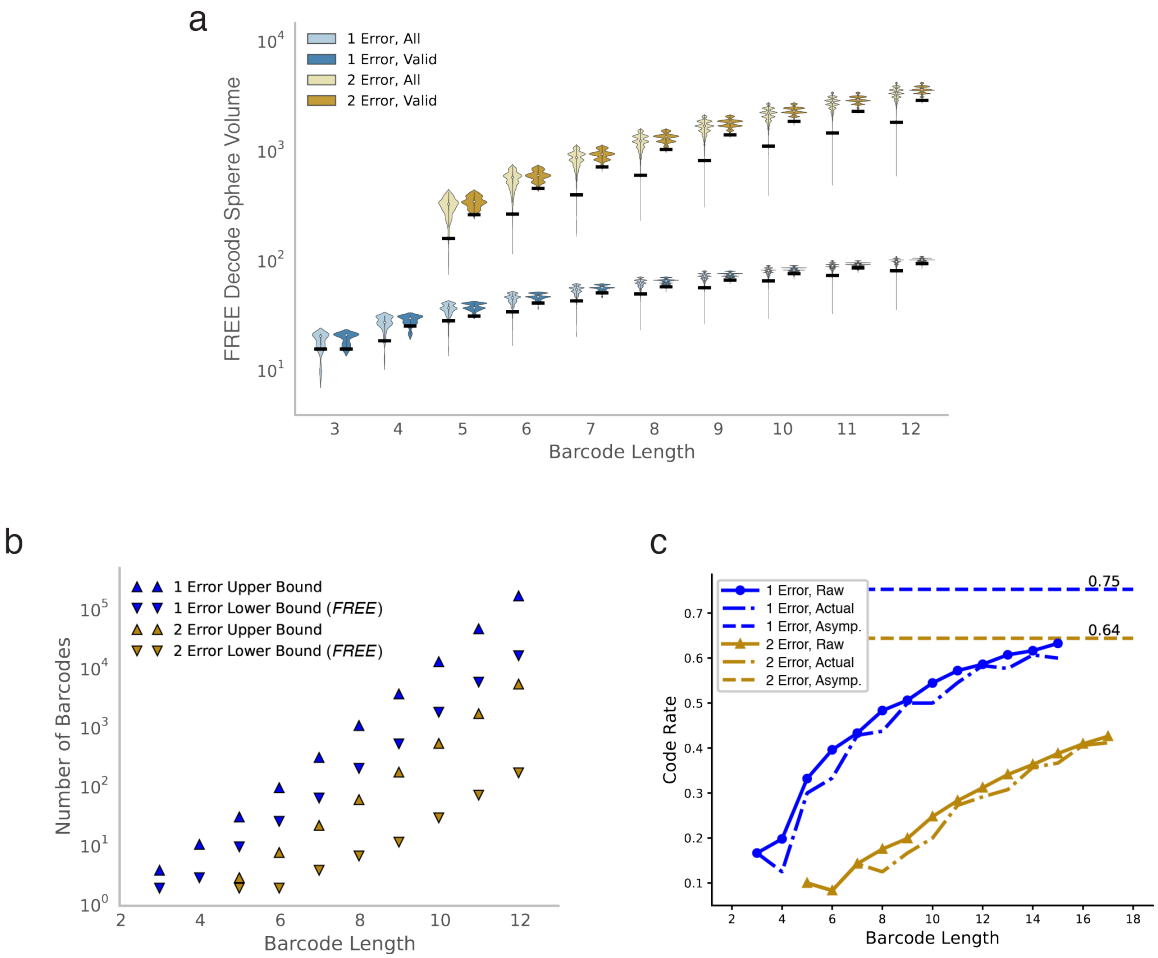


Figure 4: Decode sphere volumes and code efficiency.

a. Sphere volumes of 1- and 2-error codes for all words and for only valid code words after our FREE code synthesis and sequencing filters (no homopolymer runs, no triplet complementarity, etc.). Black lines show maximum sphere volume used for sphere packing upper bounds in (b). b. Optimal sphere packing bounds. We show the upper bound calculated for valid code words, as described in the text. The lower bound is the best efficiency achieved by any code generation method to date, which for FREE codes is simply the number of barcodes reported in this paper. c. Raw and actual code rates for each FREE barcode set included with this paper as well as the asymptotic values they approach.

Error Correction in Real and Simulated Data

We validated FREE barcodes generated in this study by both numerical simulation and experiment. Pooled oligonucleotide synthesis was used to produce a library of >8,000

oligos with double-error correcting barcodes at both ends (Figure 5a). The barcodes were arranged such that each left barcode should only ever be observed on the same oligo with one specific right barcode sequence, and similarly for right barcodes. Hence, we were able to measure the rate of incorrectly decoding barcodes from observing unexpected left-right barcode pairs (Methods). We sequenced 1.4 million copies of this library on an Illumina MiSeq for an average coverage of 159x using the standard Illumina workflow.

Full-length, paired-end Illumina sequencing was used to measure the background synthesis and sequencing error rates (Figure 5b-c). Using full-length paired-end reads permitted discrimination between synthesis and sequencing errors (Methods). Substitution, insertion, and deletion error rates from library amplification using Q5 polymerase have previously been reported to occur at rates less than 10^{-5}, and thus are a negligible fraction of the measured synthesis errors[45]. Measured errors were dominated by single-base synthesis deletions, which occured at rates of approximately 1 in 200 bp and 1 in 100 bp in the left and right barcode regions respectively (Figure 5b and Figure 29). The two-fold difference in synthesis error rates between the two sides is consistent with statements from the manufacturer regarding their synthesis error rates[32]. Sequencing error rates are between 10^{-4} and 10^{-3}, as advertised by Illumina (Figure 5c). In sum, experimental error rates are dominated by deletion errors. As Hamming codes are not designed to error-correct deletions in barcodes, they will perform very poorly in DNA-based experiments.

We compared the experimentally-determined error rates to simulations of the overall decoding error rate, i.e., the probability of incorrectly demultiplexing a barcode. Simulations were used to analyze the decode error rate for several error-correcting codes as a function of the per-base error rate, p_{err} (Figure 6). Simulations were performed in two different ways. First, we used a binomial model, which assumes independent and identically distributed errors at each base, to calculate the probability of observing more than 1- or 2-errors given per-base p_{err}. Second, we directly simulated the errors directly

using our decoding software: for a given per-base p_{err}, we randomly select barcodes and add errors with probability p_{err}. For simplicity, we model insertion, deletion, and substitution error rates of p_{err} /3 with no correlation between individual errors within a given barcode. The corrupted barcodes are then decoded using our software and the fraction of incorrectly decoded barcodes is used as a measure of the decode error rate.

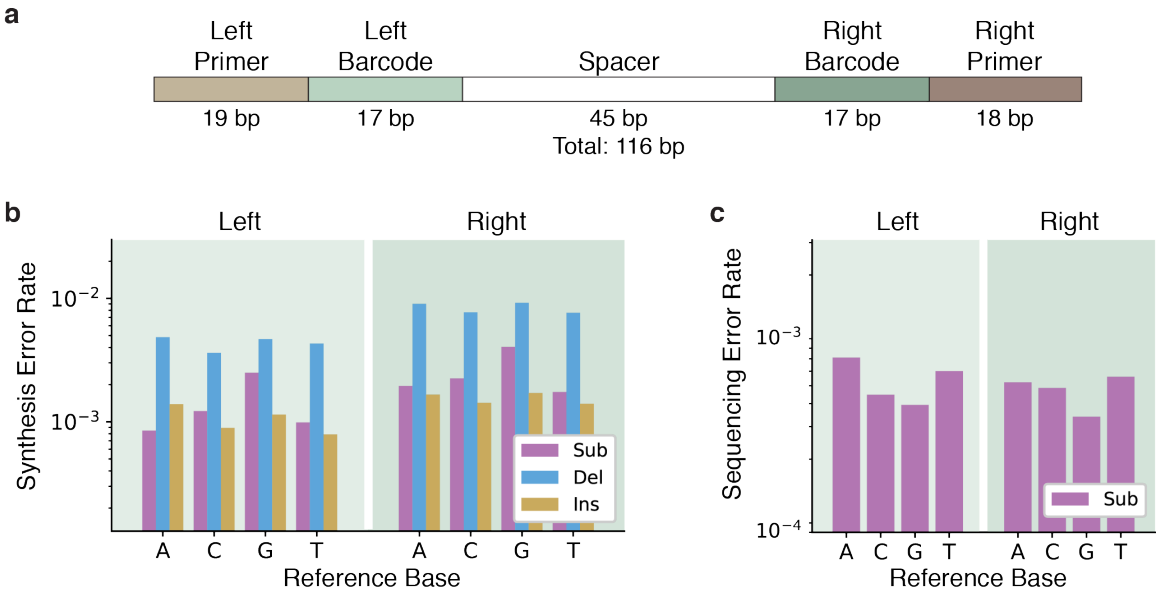


Figure 5: Experimental measurement of synthesis and sequencing error rates.

a. Schematic of the DNA constructs used for barcode validation experiments. Each member in the synthetic library had a unique pair of left and right barcodes (green) drawn from a list of >8,000 17-nt FREE codes with double-error correction. By using the primer regions (brown) to distinguish the left and right ends from one another, we could determine whether the barcodes were correctly decoded (matching) or incorrectly decoded (mismatching). b. Synthesis error rates measured in this experiment, by intended reference base and error type—substitution (sub), deletion (del), and insertion (ins). c. Measured sequencing substitution error rates, by reference base. Insertions and deletions from Illumina sequencing are extremely rare and are omitted for clarity.

At experimentally-determined per-base error rates, p_{err}, each increase in error correction level results in at least an order of magnitude improvement in the decoding error rate (Figure 6). For example, our experimental data showed an overall per-base p_{err} of approximately 10^{-2} (Figure 5b-c). At this per-base error rate, the approximate uncorrected decode error rate (solid line) is 8% for length 8 barcodes and 15% for length 16 barcodes. Without error correction a best-case scenario would be that these errors could be successfully filtered out, representing a significant loss of data. In other scenarios, these data might be erroneously counted. For zero-, single-, and double-error correction length 8 barcodes, the approximate decode error rate decreases from 8% to 0.3% to 0.005%. For length 16 barcodes, the approximate decode error rate decreases from 15% to 1% to 0.05%. A more comprehensive comparison of the various barcode lists is given in Figure 25-Figure 27. The simulated results are consistently better than the binomial approximation because indels near the right end occasionally add the correct base and because insertions occasionally push other errors out of the barcode window (Figure 24).

We validated FREE barcodes by measuring the decoding error rates for the experimental dataset described earlier (Figure 7). For double-error correction, we used mismatches in barcode pairs to identify erroneously decoded barcodes (Methods). After corrections, we observe error rates of 0.29% and 0.46% for left and right barcodes respectively. We counted the 0- and 1-error correction rates shown in Figure 7 by also counting the number of errors observed in each correctly decoded barcode. That is, 0-error correction decode error rates were calculated as the number of erroneously decoded barcodes plus the number of correctly decoded barcodes with 1 or 2 errors; 1-error correction errors were counted similarly. On the other hand, the theoretical model was calculated using the synthesis and sequencing error rates found in Figure 5 to calculate the decode error probability of each barcode depending on its base composition, and then combined for an overall error rate (Appendix C).

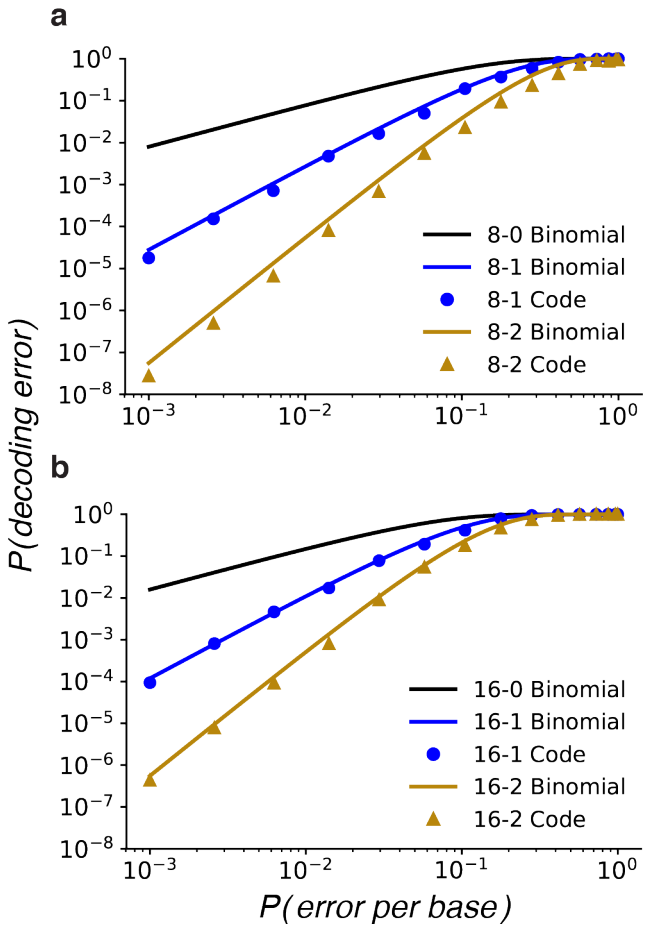


Figure 6: Decoding corrupted barcodes from simulated errors.

Modeled and simulated decoding error rates given per-base error rate for length 8 (a) and length 16 (b) barcodes. Barcode sets are labeled according to length and number of errors corrected; for example, the 16-2 code is length 16 and corrects up to 2 errors. Solid lines show the error rate approximations using a binomial model. Circles and triangles show direct simulation error rates for single- and double-error correcting codes, respectively. Substitution, insertion, and deletion errors each have simulated error rate P(error per base)/3 for simplicity.

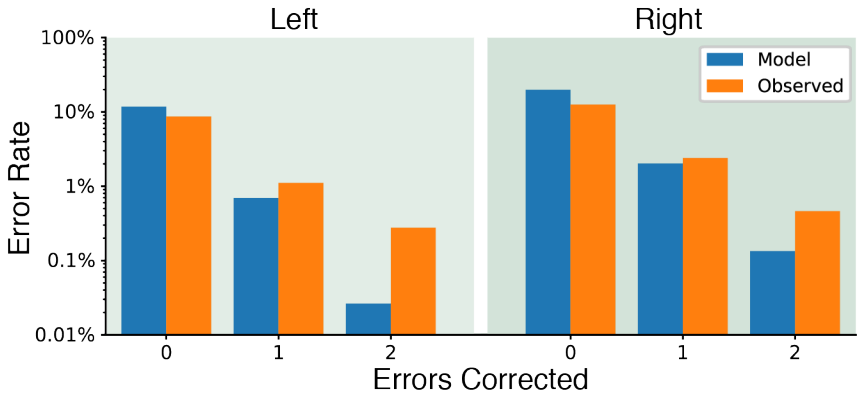


Figure 7: Decoding corrupted barcodes from experimental data.

Observed decoding error rates compared with theoretical rates from the synthesis and sequencing error rates.

The experimentally-observed decoding error rates follow the same trend as the simulated errors: decode error rates decrease by approximately an order of magnitude with each additional error-correction level. We also observed that experimental error rates are higher than the theoretical error rate. This is explained by two observations. First, the theoretical model assumes independent errors at each position along the barcode. This assumption is not observed in the experimental data (Figure 29). Second, the starting position of each barcode may not be defined exactly because the primer region can have errors. We are careful to identify the start of each barcode as precisely as possible (Appendix C), but any errors in starting position appear as spurious insertions or deletions during decoding. Nonetheless, even though per-base errors are not independent, the overall order-of-magnitude decrease in decode errors per error-correction level is recapitulated in the experimental dataset.

Combinatorially large barcode lists via concatenation

State-of-the-art high-throughput sequencing applications already require $>10^6$ unique barcodes[31]. We anticipate that improvements in high-density pooled oligo synthesis, along with the continuing reduction in sequencing costs, will continue to push the need for even

larger error-correcting barcode sets. Below, we demonstrate that arbitrarily large barcode lists (>10^{15} unique members shown here) can be constructed from FREE barcodes by concatenating multiple FREE barcodes in a row.

As a demonstration, we concatenated two or three barcodes from the same starting list of sub-barcodes (Figure 8). For the rest of this section we will refer to the original barcodes as *sub-barcodes*, while *barcode* will refer to the full length, concatenated barcode. Due to the possibility of insertions and deletions, the starting positions of the second and third sub-barcodes are only known approximately, and that approximation worsens as more sub-barcodes are added (Figure 8a). Decoding the sub-barcodes sequentially from left-to-right is a strategy to account for this ambiguity. The left-most sub-barcode is decoded first, and then the decoded sub-barcode is used to find the starting position of the next sub-barcode. The error-correction level of each FREE sub-barcode remain the same, such that, for example, three concatenated double-error correction sub-barcodes can each correct up to two errors for a maximum total of six corrected errors if and only if the errors are evenly distributed, two per sub-barcode. Overall concatenated barcode decoding error rates are given by the probability of any decoding error in any sub-barcode or -barcodes. Concatenated barcode error rates are thus slightly higher than for the individual sub-barcodes (Figure 8b). The decoding process is performed automatically using the software accompanying this paper.

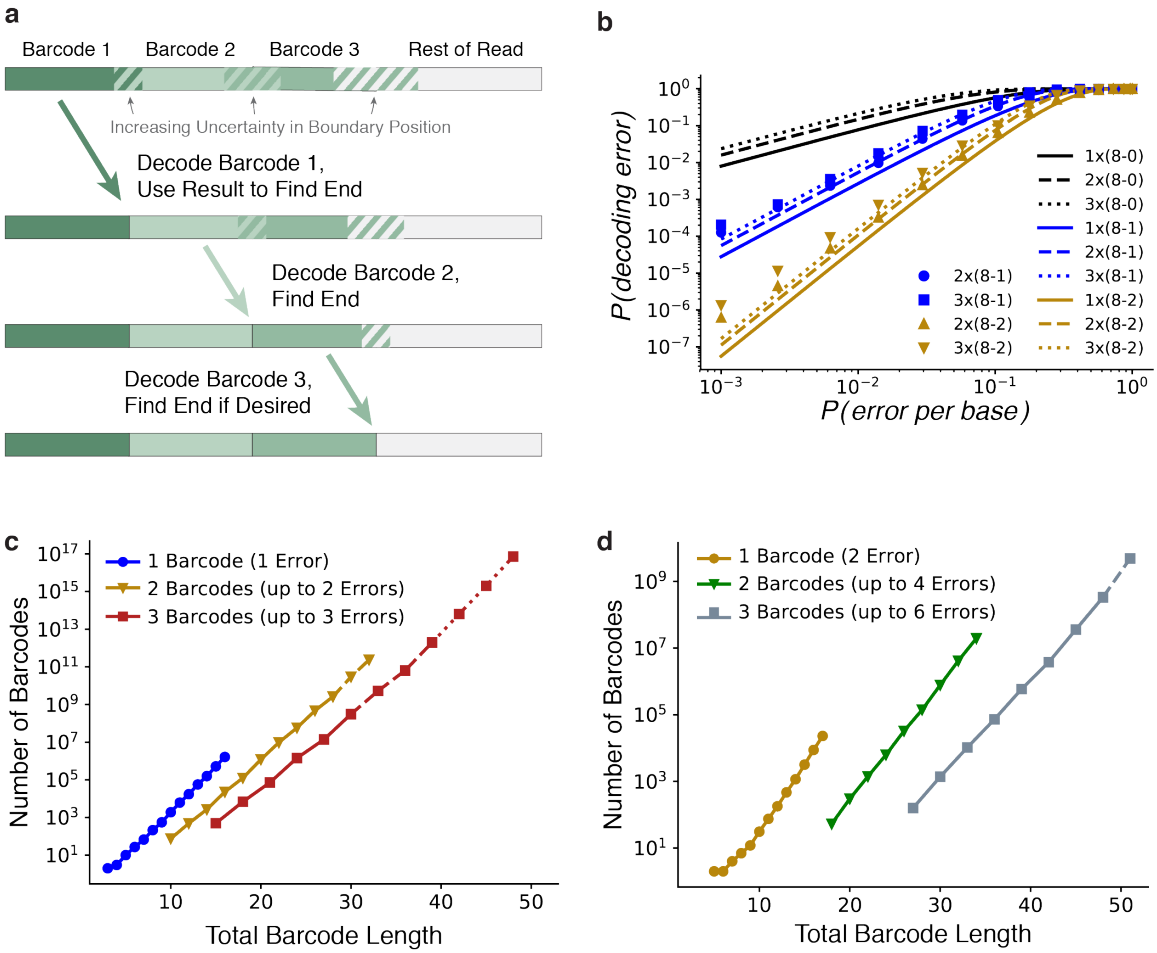


Figure 8: Combinatorial barcode libraries via concatenation of FREE barcodes.

a. Concatenated barcodes can be decoded sequentially in a left-to-right order, even when the end position of each edited sub-barcode is not initially known. The decoded first FREE sub-barcode can be used to find the starting position of the next sub-barcode, and similarly for subsequent sub-barcodes. b. Concatenated barcode decoding error rates. Concatenated barcode labels use the following format: a 3x(16-1) barcode consists of three concatenated sub-barcodes, each of which is 16 bp long and can correct up to 1 error. c, d. Concatenating multiple barcodes combinatorially increases the numbers of effective FREE barcodes. Concatenated barcodes can correct the same number of errors per sub-barcode. When the errors are distributed evenly among the sub-barcodes, concatenated barcodes can correct a higher total number of errors than the individual sub-barcodes. (c) Concatenated single-error correcting barcodes. (d) Concatenated double-error correcting barcodes. Dashed lines: projected quantities calculated by sampling; dotted lines: log-linear projections.

Concatenating FREE barcodes results in combinatorially-large barcode sets that will be sufficient for even the most demanding high-throughput sequencing applications (Figure 8). The concatenated barcodes were pruned to remain compatible with experimental constrains by removing DNA sequences that had triplet repeats of a single base or excess self-complementary (defined as any self-complementarity of any three or more bases). Even with these filters, we generated full lists of up to 10^{10} barcodes with concatenation of three single-error correcting codes (Figure 8). Beyond that, where possible, the projected total barcode count was estimated via subsampling. When even that was limited by available hard drive space, the projected total was estimated via log-linear fit, which went above 10^{15} barcodes for 3 x (16 bp single-error) barcodes. Due to their size, we do not include these concatenated barcode sets explicitly with this paper. They can be generated on demand using the included software package and single barcode lists. In sum, concatenating FREE codes produces a rapid and efficient strategy for further increasing the size of error-correcting barcode lists for pooled high-throughput sequencing experiments.

DISCUSSION

Here, we described the design and experimental validation of Filled/truncated Right End Edit (FREE) error-correcting DNA barcodes capable of correcting substitution, insertion, and deletion errors, even when the corrupted length of the barcode is unknown. We generated lists of FREE Divergence error correcting barcodes and provided software on GitHub for user-friendly generation and decoding of these DNA barcodes for real-world applications.

Most high-throughput DNA sequencing applications require PCR-based amplification or reverse transcription (in the case of RNA) of the input nucleic acid libraries. The polymerase and reverse transcriptase enzymes used during library preparation perform best on libraries that avoid stable secondary structures and self-complimentary regions. To

improve the utility of our codes for such demanding applications, we used UNAFold to calculate the melting temperature of hairpins for the FREE barcodes included with this paper[46]. This information will allow users to prune out barcode sequences with a propensity to form stable hairpins in their specific experimental conditions (Figure 30). Such experimental considerations will further increase the utility of FREE codes for demanding high-throughput sequencing applications.

In validating the FREE barcodes, we measured the types and frequency of errors that are introduced during massively-parallel oligo synthesis and Illumina-based high-throughput sequencing. We observed that deletions during synthesis were the most frequent sources of error (~1 per 100 nucleotides), followed by substitutions and insertions (~1 per 1000 nucleotides). These experimentally measured error frequencies were used to simulate and experimentally measure the decoding quality of FREE codes. Even though the observed decoding error rates do not follow a model that assumes independent errors at each base, we still obtain exponential improvement of the final decoding error rate with codes that correct for increasing numbers of errors. Importantly, the error-correcting decode software runs fast enough to handle the massive data sets involved in modern high-throughput sequencing applications, decoding hundreds of thousands of barcodes per second on a single processor for all barcode lists considered.

While we have here focused exclusively on filled/truncated *right* end edit (FREE) codes prepended to the start of sequenced DNA reads, the current work applies equally to their natural mirrored counterpart, filled/truncated *left* end edit (FLEE) codes. This would be required for applications where the barcode appears at the end of each sequenced read rather than the beginning. In fact, the same codes can be used by simply taking the reverse complement of FREE codes before synthesis and again before decoding. Hence, FREE barcodes can be used equally well on the 5' or 3' end of pooled samples, as long as the orientation is chosen appropriately.

FREE barcodes are a powerful tool to correct DNA barcode errors, reducing measurement errors in modern, high-throughput experiments. We anticipate that the use of FREE barcodes will improve these assays in three key ways: (1) helping avoid spurious results; (2) decreasing the amount of discarded data; and (3) increasing experimental signal-to-noise ratios. Decreasing spurious results and discarded data are important for any experiment involving DNA barcodes, but we are most excited by the new possibilities available with increased signal-to-noise ratios. The power to decrease error rates from 15% to 0.05%, as in Figure 6b, could open the door for entirely new assay designs. We anticipate that FREE barcodes will be broadly useful for the ever-growing set of pooled high-throughput sequencing experiments in cell and molecular biology, protein engineering, and drug discovery.

METHODS

Definitions and Numerical Representation of DNA

For any barcode system, the word length, *n,* is given. Any DNA sequence of length *n* is a *word*, and any word observed in the data is an *observed word*.

We represent strings of DNA as base-4 numbers where A, C, G, and T correspond to 0, 1, 2, and 3 respectively. So, for example,

$$AAGCT = (00213)_{base\,4} = 39 \ \ length\ 5$$

Here 39 is the *word number* and 5 is the *word length*. Note that the word length is required to uniquely convert numbers to DNA to account for leading A's. For example, the word number from the example above, 39, with word length 3 is simply GCT. For word length *n*, the largest valid word number is $4^n - 1$.

For an *m*-error correcting code we define a *decode sphere* around a barcode *B* to be the set of all words with FreeDiv less than or equal to *m,* and we define an *encode sphere* to be the set of all words of FreeDiv less than or equal to 2*m*. We write these as *DecodeSphere(B)* and *EncodeSphere(B)*.

Barcode Generation

FREE barcode sets are generated with a modified lexicographic code generation method. Lexicographic code generation consists of marching through all words lexicographically, alphabetically in this case, and adding new words to the list of barcodes whenever they are sufficiently far from all previous barcodes[47]. For Hamming codes, lexicographic codes are linear[47], and more generally, lexicographic code generation has been shown to have relatively good sphere packing efficiency[41]. The first FREE modification to the procedure is to enforce the following sequencing and synthesis properties:

- Balanced GC content (40-60%)
- No homopolymer triples (e.g., TTT)
- No triplet self-complementarity
- No GGC (Illumina error motif[42])

For speed we iterate over these potential barcodes via recursive base addition: given a barcode prefix *P*, we add the next base only if it does not violate any of the above. We thereby skip large recursive subtrees in which all words violate one of the above conditions.

For an *m*-error correcting code, the only requirement is that the decode spheres of all barcodes are disjoint. Because FREE divergence is not a metric, standard metric-based code generation methods cannot be used. Instead, we accomplish this directly with a sphere iterator (Appendix C). For every accepted barcode *B*, we iterate over *DecodeSphere(B)* and reserve all words therein as mapping to *B*. And for any potential new barcode *P*, we first verify no words in *DecodeSphere(P)* are reserved before accepting it as a new barcode.

This algorithm would be very slow because most decode sphere tests would run into reserved words and fail to add new barcodes. One further observation makes this process tractable. Given a barcode B and a proposed new barcode *W,* if $FreeDiv(B, W) \leq 2m$, that is, if *W* is in *EncodeSphere(B)*, then *DecodeSphere(W)* and *DecodeSphere(B)* overlap and *W* is not a valid new barcode. This implies the following algorithm: generate the code by lexicographically iterating over words while looking for new barcodes to add to the code. For each accepted new barcode *B*, we color any uncolored words in *EncodeSphere(B)* black, and then we color all words in *DecodeSphere(B)* red. Restricting encode sphere coloring to previously uncolored words avoids overwriting the decode spheres of all previous barcodes. All black- and red-colored words are guaranteed to not be valid barcodes, so addition of new barcodes is restricted to uncolored words. For an uncolored proposed new barcode *W*, *DecodeSphere(W)* is checked for red words. If no red words are found, *W* is added as a new barcode.

The coloring of barcodes, decode spheres, and encode spheres is accomplished by having an array of 4^n integers valued 0, 1, or 2: 0 for uncolored, 1 for black, and 2 for red. The location of each integer in memory itself represents the word, via the numerical representation of DNA given above. This is both memory and speed efficient. Memory efficiency is important, as it is a limiting resource for this method. The memory required for barcode generation is 4^k bytes, which for this paper was up to 16Gb of random access memory (RAM).

Barcode Decoding

The decoding process builds the code book and looks up decoded words directly. We do this in a memory efficient fashion as follows. For each barcode in a list, the *barcode index* is defined as the index of that barcode within the list of barcodes. We again reserve a space of 4^k integers to represent the code space. For each barcode *B*, we store the barcode index of *B* at every word of *DecodeSphere(B)*. We store barcode indices rather than barcode

numbers because barcode indices require fewer bits per word. The memory required for barcode decoding is $(1, 2, or\ 4) \times 4^n$ bytes, requiring 1, 2, or 4 bytes to store each barcode index. For this paper, the maximum memory used for barcode decoding was 32Gb of RAM.

Barcode Pruning

Specific barcode lists from literature or elsewhere may sometimes be required for a given experiment, but require pruning to find a subset with error-correction. We accomplish barcode pruning via the same strategy as barcode generation, but only considering the input set of barcodes as potential new barcodes. This pruning method was also used to prune the linear Hamming codes.

Simulation of Errors

To test the error-correcting capacity of FREE barcodes, we wrote error-simulating code which adds a given number of substitutions, insertions, deletions, or all three randomly distributed. We used this to verify the correctness of each of the FREE *m*-error correcting codes by randomly selecting barcodes, adding *m* errors, and verifying that the decoded word matches the expected word. We used the same code for generating Figure 6 by randomly choosing the number of errors from a binomial distribution with probability of error p_{err}.

Levenshtein Barcodes

Levenshtein barcodes were generated lexicographically using the standard technique of code generation with a metric. Briefly, for desired barcode length *n* and number of correctable errors *e*, we walk through the space of *n*-mers lexicographically adding any new word if it: (a) satisfies the same sequencing and synthesis properties as above, and (b) is Levenshtein distance at least 2*e*+1 from any previously accepted barcode.

Experimental Synthesis, Sequencing, and Decoding Error Rates

Oligonucleotide pools were designed as in Figure 5a, with primers and barcodes on each end and a spacer in the middle (116 bp total length). To test the FREE method, 8,634 barcodes of length 17 and double-error correction were used in 8,634 unique pairs. Oligos were synthesized (CustomArray), and the oligo pool was amplified for twenty cycles with Q5 polymerase (NEB) and sequenced on an Illumina MiSeq machine with 2x150 bp paired-end reads. Maximum likelihood sequences were inferred using both reads.

The left and right primer sequences were used to determine both the read orientation and the starting position of each barcode (Appendix C). Each barcode was then decoded using the FREE decoding software. Matching barcodes identified correctly decoded barcodes, while mismatching barcodes indicated an error. The FREE method was powerful enough to reveal a surprising and unrelated source of error: the creation of oligo chimeras, sequences with the left part of one oligo and right part of another, which we then also accounted for (Appendix C).

Once each oligo had been identified from its barcodes, the observed sequence was aligned with the reference sequence. At each base where the two reads agreed with each other but not with the reference sequence we counted a synthesis error, at each base where the reads disagreed and one read matched the reference sequence we counted a sequencing error, and at each base where the reads disagreed and neither matched the reference sequence we counted a synthesis and a sequencing error.

Observed synthesis and sequencing error rates for each reference base were used to find theoretical decoding error rates for each barcode given its base composition. These were then used to estimate overall expected error rate.

Decode error rate model

The decoding error rate of an m-error correction code is the probability of seeing more than m errors in a given barcode. For error analysis, we model each barcode as a queue of intended bases. At each read position, an intended base is popped off the queue and attempted to be added. One of four things will happen: 1) the correct base will be added, 2) an incorrect base will be added, 3) the base will be deleted, or 4) another base will be inserted and the intended base will go back to the top of the queue. The first three options do not return the base to the queue, resulting in the same structure of expected output 7→ observed output. However, insertions cause the intended base to return to the top of the queue, and the output was never expected in the first place. For this reason, it must be modeled differently from the other three. Assuming independent errors of all types and positions, we model insertions with a negative binomial distribution and the correct bases, deletions, and substitutions with a multinomial distribution, using our measured error rates per reference base, shown in Figure 5.

Let a barcode be given and let B be the 1-by-4 row vector with counts of each of the bases ACGT in the given barcode. Let I be the 1-by-4 row vector of insertion counts for each of the four bases. Further, let CDS be the 3-by-4 matrix with columns corresponding to the DNA bases, and rows corresponding to all non-insertion outputs: correct bases, deletions, and substitutions. We will occasionally refer to the rows of CDS individually as C, D, and S, but we leave it in matrix form as they are tightly connected. In fact, it must be true that $C + D + S = B$.

We use the measured error rates given reference base shown in Figure 5. Insertion and deletion rates, $p_i(b)$ and $p_d(b)$, are taken directly from synthesis error rate measurements. Substitution rates, $p_s(b)$, are calculated as the probability of not observing the event {no synthesis substitution and no sequencing substitution} nor the event {synthesis substitution to another base c and correcting synthesis substitution back to b}, and are thus given by

$$p_s(b) = 1 - (1 - p_{s,synth}(b))(1 - p_{s,seq}(b)) \; - \sum_{\substack{c \in \{ACGT\} \\ c \neq b}} \frac{1}{3} p_{s,synth}(b) \cdot \frac{1}{3} p_{s,seq}(c)$$

Now let $p_c(b) = 1 - p_d(b) - p_s(b)$ be the probability of correctly adding a base, let N be the random variable for the total number of errors, let n_{err} be given, and let n_i, n_d, and n_s be the number of insertions, deletions, and substitutions respectively. Then, from our assumption of independent error rates given reference base, we get for each reference base the previously mentioned negative binomial distribution for insertions and multinomial distribution for the rest:

$$p(N = n_{err} \mid B) = \sum_{\substack{CDS \text{ with } n_d, n_s \\ I \text{ with } n_i \\ n_d + n_s + n_i = n_{err}}} \prod_{b \in \{ACGT\}} \left[\binom{I_b + B_b - 1}{I_b} (1 - p_i(b))^{B_b} \, p_i(b)^{I_b} \right.$$
$$\left. \times \binom{B_b}{C_b + D_b + S_b} p_c(b)^{C_b} p_d(b)^{D_b} p_s(b)^{S_b} \right]$$

Finally, we marginalize the above over barcode identity,

$$p(N = n_{err}) = \sum_{B \in Barcodes} p(N = n_{err} \mid B) \, p(B)$$

and sum over n_{err} as required.

Chapter 2: CHAMP: A Massively Parallel Protein-DNA Interaction Mapping Platform†

INTRODUCTION

CRISPR systems, composed of clustered regularly interspaced palindromic repeats (CRISPR) of DNA at specific CRISPR genomic loci and a number of CRISPR-associated (Cas) proteins, provide bacteria and archaea with adaptive immunity against invading phages and other foreign nucleic acids[49,50]. To provide adaptive immunity, cells assemble a CRISPR RNA (crRNA) and a Cas protein or proteins into an RNA-guided nucleoprotein complex that recognizes specific foreign DNA targets. After target DNA recognition, a CRISPR-specific effector protein degrades the foreign nucleic acids. CRISPR systems also confer immunity against future infections by acquiring foreign DNA sequences and inserting them as the eponymous spacers (between palindromic repeats) in the CRISPR locus[51]. Recent breakthroughs with diverse CRISPR-Cas systems have enabled microbiologists to program DNA/RNA targeting, leveraging this microbial immune strategy for diverse biotechnological and medical applications[52,53].

Intense interest in emerging CRISPR-Cas systems has driven the development of high-throughput methods for characterizing crRNA-guided binding and cleavage activities. Current strategies typically use deep sequencing to identify off-target binding (e.g., ChIP-Seq, pulling down short stretches of protein-bound DNA with protein-specific antibodies

---

† This chapter draws on material from Jung C, Hawkins JA, Jones SK, Xiao Y, Rybarski JR, Dillard KE, Hussmann J, Saifuddin FA, Savran CA, Ellington AD, Ke A, Press WH, and Finkelstein IJ. Massively Parallel Biophysical Analysis of CRISPR-Cas Complexes on Next Generation Sequencing Chips. Cell 170, 35–47.e13 (2017).[48] J.A.H., C.J., S.K.J., J.R.R., C.S., W.H.P., and I.J.F. designed research. J.A.H., C.J., S.K.J., Y.X., J.R.R., K.D., M.A.S., and C.S. performed research. J.A.H., J.H., and J.R.R. wrote the software. J.A.H., C.J., S.K.J., Y.X., and J.R.R. analyzed the data. J.A.H., C.J., S.K.J., W.H.P., and I.J.F. wrote the paper. All authors commented on the manuscript.

and then deep sequencing) and cleavage (e.g., Digenome-Seq, digesting genomic DNA with a nuclease of interest and then deep sequencing)[54–59]. Alternative strategies include *in vivo* fluorescent reporters for CRISPR-Cas protein binding or for the repair of resulting DNA double strand breaks[55,58,60,61]. These methods frequently detect off-target binding and cleavage activities, but also have several limitations[62]. For example, readouts such as green fluorescent protein (GFP) production or DNA break repair may vary with cell cycle stage and genomic context. Similarly, pulldown methods can be influenced by antibody quality, the degree of chemical crosslinking, and the chromatin state of a given target. Most of these strategies are also limited to identifying genomic off-target DNA cleavage sites, thereby making it difficult to place the results in a quantitative biophysical framework. In short, methods that aim to identify off-target sites *in vivo* are not optimal for probing the molecular mechanisms underlying CRISPR-Cas activities.

Here, I describe a chip-hybridized affinity-mapping platform (CHAMP) for comprehensively profiling protein-nucleic acid interactions on already-sequenced next generation sequencing (NGS) chips, which I developed in collaboration with an experimental group (Dr. Cheulhee Jung and Dr. Stephen K. Jones Jr., in Dr. Finkelstein's lab). The most widely adopted NGS sequencers fluorescently image clusters of DNA molecules covalently affixed to the surface of a microfluidic chip. CHAMP leverages these chips—normally discarded after sequencing—to quantitatively measure protein-DNA interactions. Importantly, CHAMP does not require any hardware or software modifications to older NGS sequencers, as has been reported previously[63–65]. Instead, it uses the modern and ubiquitous Illumina MiSeq instrument to generate chips and sequencing data. Protein-DNA profiling experiments are then performed independently on a standard fluorescence microscope. In short, NGS sequencing provides information about the position and identities of millions of different DNA molecules, while the microscopy experiments quantitatively measure the apparent binding affinity of the proteins to these DNA sequences.

We used CHAMP to quantitatively profile interactions between the *T. fusca* Type I-E CRISPR-Cas (Cascade) effector complex and a diverse library of target DNA molecules. The Type I system accounts for approximately 50% of bacterial CRISPR systems, and has been used to control gene expression and cell fate[66–69]. Using CHAMP, we profiled three aspects of Cascade-DNA interaction: protospacer adjacent motif (PAM) recognition, tolerance of mismatches in the target sequence, and recruitment of the Cas3 nuclease subunit.

First, we considered recognition of the protospacer-adjacent motif (PAM). In all CRISPR-Cas systems, the PAM flanks target DNA that is complementary to the crRNA and is recognized in a sequence-specific fashion by the protein alone. The PAM is crucial for facilitating interrogation of the target DNA by the Cascade complex. Diverse PAMs can also bias CRISPR-Cas systems towards DNA degradation (interference) or spacer acquisition (adaptive immunity)[70–74]. Early studies proposed that Cascade recognizes a three nucleotide PAM on the 5' end of the target[74–76]. However, recent structural and sequencing studies of the *E. coli* Cascade complex suggested that the Cse1 subcomplex is sensitive to an extended PAM[69,77]. CHAMP profiling of a synthetic nucleotide library demonstrated that Cascade indeed recognizes an extended, six-nucleotide PAM.

Second, CHAMP profiling of DNA-binding against sequences with mismatches in the target sequence reveals a three-nucleotide periodicity of decreased specificity in Cascade-DNA interactions, as well as an overall decline in sequence-specificity with distance from the PAM.

Finally, using a three-color experiment, we demonstrated that recruitment of the nuclease subunit Cas3 is sensitive to the identity of the PAM sequence and PAM-proximal DNA-RNA mismatches, providing evidence for a novel DNA-guided proofreading mechanism.

These results accurately reproduced *in vivo* interference experiments, reflecting the strength of CHAMP for mapping protein-DNA interactions. More broadly, this study provides an experimental and computational framework for comprehensive analysis of protein-DNA interactions for diverse CRISPR systems, RNA-guided nucleases, and other DNA-binding proteins.

RESULTS

A chip-hybridized affinity-mapping platform (CHAMP) for profiling CRISPR-Cas DNA interactions

The CHAMP assay is conceptually quite simple (Figure 9A). First, we generate a DNA library of interest and sequence it on an Illumina NGS sequencer, a MiSeq sequencer in the present work. At the end of the DNA sequencing run, the surfaces of the Illumina NGS chips are decorated with millions of spatially registered, unique DNA clusters of known sequence. Using a total internal reflection fluorescence (TIRF) microscope, we then image protein-DNA binding with fluorescently labeled proteins at a series of increasing concentrations. We finally use an image alignment and analysis software pipeline to infer sequence-specific protein-binding affinity for each sequence identity in our library. CHAMP's strength lies in its platform independence and its software pipeline.

After sequencing but before flowing protein onto the chip, there are some additional steps of interest required to prepare the chip. After sequencing, the chip is covered in residual fluorescent nucleotides left over from the sequencing process that would confound imaging if left on the chip. These are specific to the DNA strand not covalently attached to the chip surface. So first, we strip away said fluorescent nucleotides and regenerate dark double-stranded DNA (dsDNA) in the ~20 million DNA clusters on the surface of a sequenced NGS chip (Figure 9A**,** Figure 31). Second, to facilitate alignment of fluorescent clusters

with their DNA sequences, we hybridize a fluorescent oligonucleotide primer to a known subset of the DNA clusters for use as an alignment marker in the downstream image-processing pipeline (Figure 9A).

The main challenge for CHAMP is the precise mapping of each fluorescent DNA cluster to an underlying DNA sequence. This information is partially encoded in the sequencing output generated by all Illumina sequencers, reported in text files called FASTQ files. However, CHAMP utilizes images obtained via conventional TIRF microscopy rather than an Illumina sequencer (Figure 9A, right). These images are transformed by an arbitrary translation, scaling, and rotation relative to the coordinate system used in the Illumina software. Alignment between the Illumina output and CHAMP images is further confounded by false-positive (e.g., spurious fluorescent signals) and false-negative cluster coordinates (e.g., fluorescent signals that are filtered out by the Illumina pipeline). To overcome this, CHAMP uses alignment markers with known DNA sequences to match the spatial position of all fluorescent clusters to a corresponding record in the sequencing output file (Figure 16A). We utilized a library made of PhiX bacteriophage genome as our alignment marker. The PhiX library is included as an internal control on every Illumina chip, and typically comprises 5-10% of all sequenced DNA clusters. This library also contains a unique adapter, which can be selectively illuminated with a fluorescent primer (Figure 31). Mapping the alignment markers and protein-bound clusters to their sequences requires two stages: first, a rough alignment using Fourier-based cross correlation methods is performed, followed by a precision alignment using what I call constellation mapping, which consists of determining constellations of FASTQ points and imaged clusters which represent the same points in the two spaces followed by least squares fitting between the two constellations (Figure 16 and Computational Methods). This is a specialized example of the image registration problem[79–81], and allows CHAMP to function with any sequencing platform and TIRF microscope.

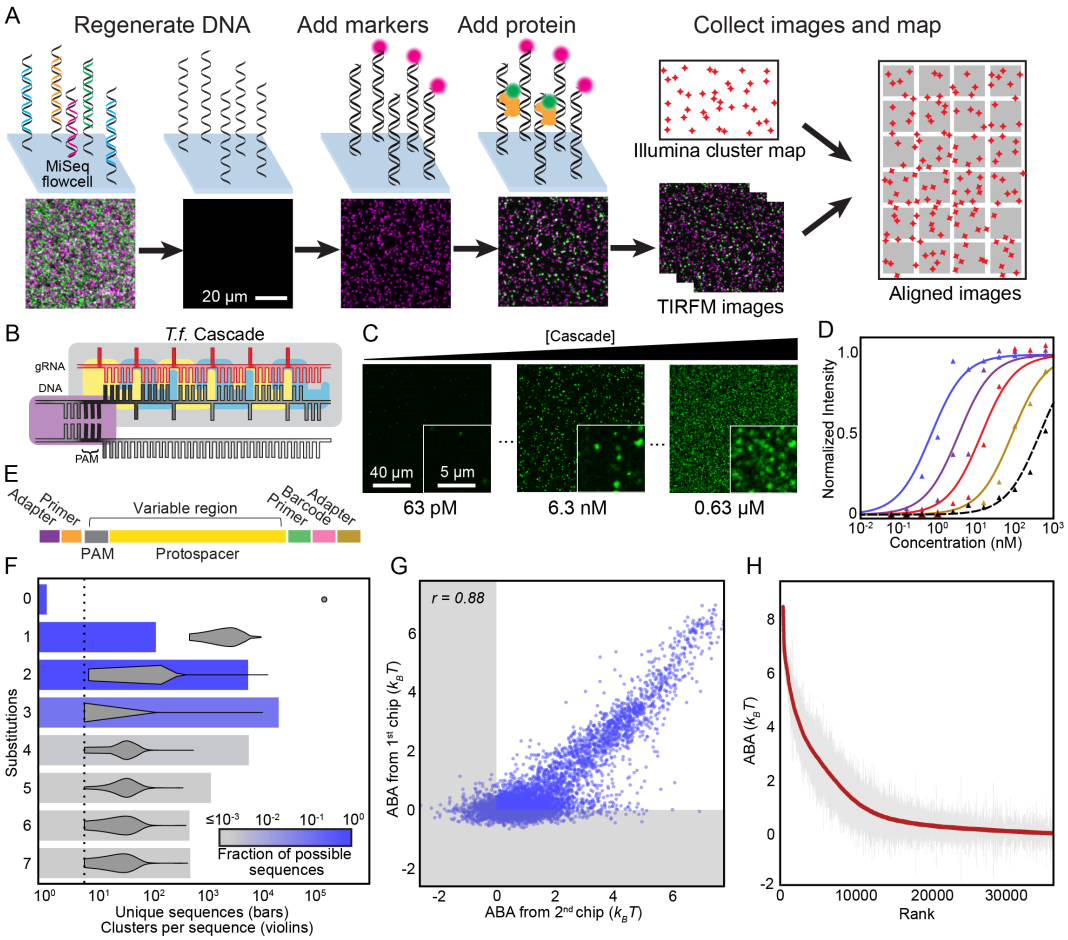


Figure 9. A chip-hybridized affinity-mapping platform (CHAMP).

Figure 9. A chip-hybridized affinity-mapping platform (CHAMP).

(A) Overview of the CHAMP workflow. DNA is regenerated on a sequenced NGS chip. A subset of clusters is hybridized to fluorescent oligonucleotides (alignment markers, magenta). Fluorescent proteins are incubated in the chip (green) and the fluorescent intensities at each DNA cluster are recorded via TIRF microscopy. A computational pipeline uses the alignment markers to identify the DNA sequences of all fluorescent clusters. (B) A schematic representation of the *T. fusca* Cascade protein complex. Cse1 is shown in purple, Cas7 subunits are shown in alternating blue and yellow, and all other subunits are collectively represented in gray. The target DNA is gray, the protospacer adjacent motif (PAM) and seed regions are black, while the crRNA is red. (C) Increasing concentrations of fluorescent Cascade complexes are incubated in the regenerated NGS chip and (D) the apparent binding affinities for each DNA sequence are obtained by fitting the fluorescent intensities to the Hill equation. Each curve represents the apparent binding affinity calculated from at least five unique DNA clusters. The lowest-affinity curve in (D) reports non-specific binding of Cascade to off-target DNA clusters. (E) Illustration of the synthetic oligonucleotide library used for CHAMP. The PAM and protospacer regions are randomized during library synthesis. (F) Overview of the randomized library used for these studies. The bar graph represents the number of unique sequences used in the CHAMP experiments with increasing substitutions from the ideal PAM and protospacer sequence. The bars are shaded to indicate the percent coverage of the relevant sequence space. Violin plots indicate the number of DNA clusters observed per sequence in the CHAMP dataset. Only sequences represented by five or more unique DNA clusters are included in the analysis (dashed line). (G) CHAMP experiments were highly repeatable between two independently sequenced NGS chips. The gray zones indicate ABAs that fell outside of our experimentally defined cutoff for non-specific binding. The r-value was calculated omitting gray zones. (H) A rank-ordered list of all 35,968 ABAs that were measured via CHAMP. The gray line represents the standard deviation as measured by bootstrap analysis[78].

Using CHAMP, we profiled the PAM specificity and off-target binding affinity of the mesophilic *T. fusca* Type I-E CRISPR-Cas (Cascade) complex (Figure 9B). Experiments were carried out on regenerated MiSeq chips that contained a synthetic oligonucleotide library encoding substitutions within the PAM and the target DNA sequence. DNA binding was imaged at eleven Cascade concentrations ranging from 63 pM to 630 nM (Appendix D). At each concentration, the mesophilic Cascade complex was first incubated in the chip at 60˚C to promote DNA binding. Next, unbound complexes were flushed out of the chip, and DNA-bound Cascade was rapidly cooled to room temperature and labeled *in situ* with fluorescent anti-FLAG antibodies (Figure 9C). The *T. fusca* Cascade complex included a triple FLAG epitope on the C-terminus of the Cas6 subunit. We confirmed that this epitope tag did not alter DNA binding by the *T. fusca* Cascade (Figure 32), as reported for the *E. coli* Cascade complex[82–85]. We did not observe significant Cascade loss or photobleaching during the course of image collection (~15 minutes per protein concentration) (Figure 33).

With this data, we then determined the apparent binding affinity, defined below, of Cascade to each sequence. First, apparent K_d values were determined by fitting the fluorescence intensities of each DNA cluster at the eleven Cascade concentrations to the Hill equation (Figure 9D and Computational Methods). Non-specific DNA binding was observed via a random DNA sequence that was also included in the chip. This negative control sequence had an apparent K_d that was lower than our highest measured concentration (Figure 9D, dashed curve). I used these fits to define apparent binding affinity (ABA), the difference in apparent ΔG between the negative control sequence and a sequence of interest (Computational Methods). Positive values indicate stronger binding, and negative values were discarded as non-specific DNA binding. DNA sequences with at least 5 unique fluorescent clusters were included in the analysis, which provided average error of approximately 0.2 k_BT for the apparent binding affinity (Figure 18).

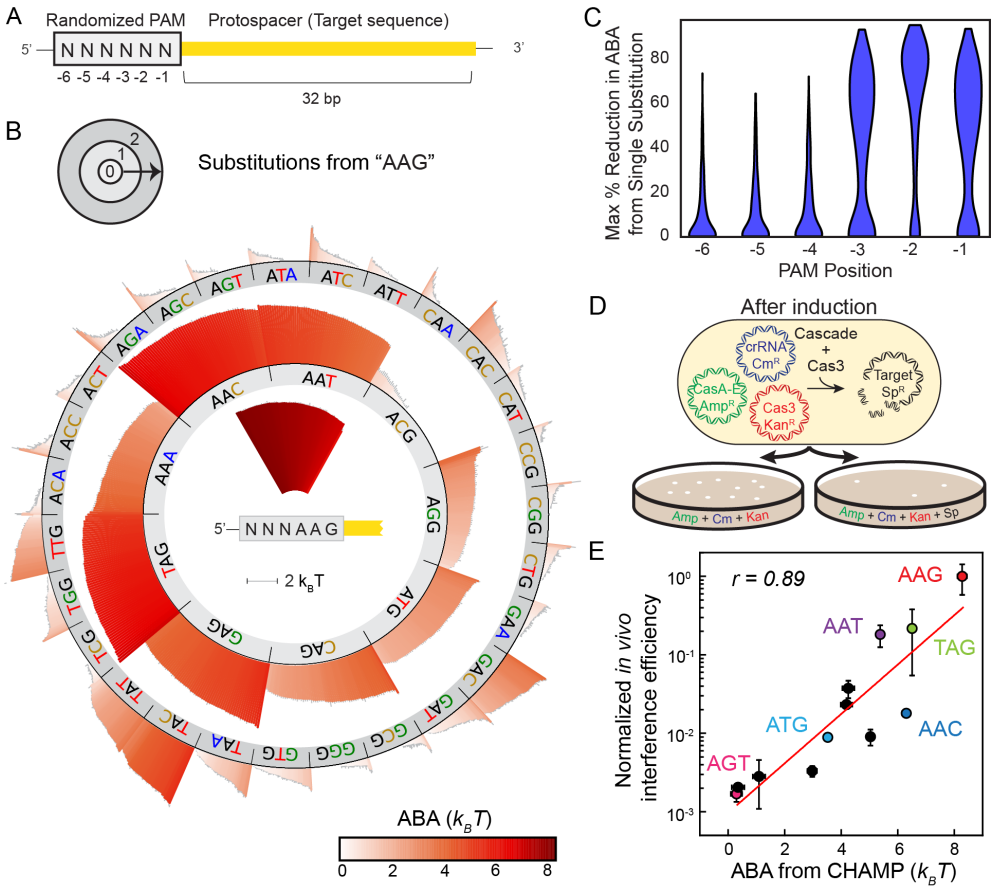


Figure 10. Cascade recognizes an extended protospacer adjacent motif (PAM).

(A) Overview of the randomized DNA library used for profiling extended PAM recognition. (B) A PAM landscape plot summarizes the ABAs for all non-zero six-nucleotide PAM sequences. The plot is organized into three concentric rings (top). These rings are organized by the sequence of the minimal, three nucleotide PAM. The inner ring represents all 64 ABAs obtained by randomizing the PAM_{-4} to PAM_{-6} positions for the strongest minimal PAM (e.g., $N_{-6}N_{-5}N_{-4}A_{-3}A_{-2}G_{-1}$). The outer rings show ABAs for all extended PAMs that are related by one or two nucleotide substitutions to the minimal $A_{-3}A_{-2}G_{-1}$ PAM. The heights of the bars and the color map represent the ABAs. (C) Maximum percent reduction in ABA due to a single substitution at a given PAM position. For each set of sequences varying only in the indicated position (other positions held constant), the difference between the maximal and minimal ABAs was calculated, adjusted to remove possible differences due to error in ABA measurements (95% confidence). Violin plots show the distribution of resulting percent reductions for all such sets of sequences. (D) Illustration of the plasmid-based *T. fusca* Cascade/Cas3 *in vivo* interference assay. Degradation of the target DNA plasmid by Cascade/Cas3 removes streptomycin resistance. (E) *In vivo* interference is strongly correlated with the ABAs measured via CHAMP. Error bars represent three biological replicates (*in vivo* assays) or the standard deviation of the ABAs determined via bootstrapping.

This CHAMP dataset resulted in ~36,000 unique DNA sequences with ABAs that were above the non-specific DNA binding threshold (Figure 9H). We sequenced ~16 million target DNA sequence clusters, giving complete coverage of all possible six-nucleotide PAM variants, as well as all single- and double-nucleotide substitutions along the entire target DNA (Figure 9E, Figure 9F). Paired-end reads of linearly amplified synthetic oligonucleotide libraries were used to minimize biases and errors from library construction, synthesis, and sequencing with maximum a posteriori base identification using a Bayesian model I developed (Figure 35). To avoid chip-specific biases, we performed experiments on two independent MiSeq chips, which recapitulated the measured ABAs (r=0.88) (Figure 9G). With this dataset, we next set out to determine the guiding principles of Cascade-DNA interactions.

Quantitative profiling of the protospacer adjacent motif (PAM)

We used CHAMP to determine the apparent binding affinity of Cascade towards six nucleotide PAMs when the target DNA is fully complementary to the corresponding crRNA (Figure 10A). CHAMP profiling of all 4,096 unique six nucleotide PAMs resulted in 950 sequences that had a positive apparent binding affinity (ABA). In order to reduce the dimensionality of this data, I adapted the graphical technique of sequence specificity landscapes to visualize the complete set of all PAM preferences (PAM landscape, Figure 10B)[86]. The PAM landscape displays all PAM-dependent ABAs as a series of concentric rings (Figure 10B, top). The highest-affinity sequence for the first three PAM positions is well documented ($A_{-3}A_{-2}G_{-1}$) and is included in the center of the concentric rings. This innermost dataset displays the ABAs for all 6-nucleotide PAM sequences that contain a perfect match to the highest affinity three-nucleotide "minimal" PAM ($N_{-6}N_{-5}N_{-4}A_{-3}A_{-2}G_{-1}$ for *T. fusca* Cascade: 64 unique sequences). The height and color of each peak on the individual rings corresponds to the ABA. A grey line above each peak represents the

standard deviation of each measurement, as determined by bootstrap analysis. The vertical bars are sorted from the highest to lowest affinity sequences for each minimal PAM. When paired with AAG, variation in the -6 to -4 position contributes minimally to the ABA. The next ring in the landscape shows ABAs for six nucleotide PAMs that vary from $A_{-3}A_{-2}G_{-1}$ by a single nucleotide in the first three positions (e.g., $N_{-6}N_{-5}N_{-4}C_{-3}A_{-2}G_{-1}$). The final ring shows PAMs that vary from $A_{-3}A_{-2}G_{-1}$ by two nucleotides (*e.g.*, $N_{-6}N_{-5}N_{-4}C_{-3}C_{-2}G_{-1}$). We did not detect any measurable binding affinity to PAMs with three substitutions relative to $A_{-3}A_{-2}G_{-1}$; this circle is not displayed in Figure 10B. This representation gives a high-level overview of the entire PAM sequence space, reducing the high-dimensionality of CHAMP datasets for rapidly comparing the binding affinity to various PAMs.

I determined the relative importance of each base in the extended PAM by computing the maximum change in the ABA when only that base was varied (Figure 10C). For example, one data point in the PAM Position -6 distribution is the maximum difference between the four sequences with PAMs of NAAAAA. These results show that the PAM_{-2} position is the most critical for defining the highest-affinity *T. fusca* PAM. In contrast, the closely-related *E. coli* Cascade complex has promiscuous recognition at the PAM_{-2} position[77] (Figure 10C). Both PAM_{-1} and PAM_{-3} make similar contributions to the ABA. Subsequent positions in the extended PAM typically contribute less to ABA ($PAM_{-2} > PAM_{-1} \approx PAM_{-3} > PAM_{-4} > PAM_{-5} > PAM_{-6}$). These results also highlight that PAMs with intermediate ABAs are the most sensitive to the identity of nucleotide positions -4 to -6. The importance of these positions is significant for PAM sequences such as <u>NNN</u>GAG, where apparent binding affinity increases over 60% from 2.7 k_BT for <u>GGA</u>GAG to 4.4 k_BT for <u>CAC</u>GAG. The PAM_{-4} position is likely decoded by direct interactions with the PAM-interacting subunit, Cse1, as reported for the *E. coli* Cascade structure[77]. Contributions of PAM_{-5} and PAM_{-6} may be due to indirect effects such as changes in the shape of the DNA minor groove.

We next compared the CHAMP results with *in vitro* electrophoretic mobility shift assays (EMSAs) and *in vivo* interference assays. EMSAs showed excellent agreement with the CHAMP datasets (r=0.96) over three orders of magnitude in concentration (Error! Reference source not found.). As expected, purified Cascade complexes lacking the PAM-interacting Cse1 subunit did not exhibit any target DNA binding via EMSAs or CHAMP. Next, I compared the results obtained via CHAMP with plasmid-based interference assays obtained for a variety of PAM sequences[87]. In this assay, *T. fusca* Cascade, along with Cas3 nuclease, is induced in cells that also harbor a target plasmid (Figure 10D). Degradation of the target plasmid yields loss of antibiotic resistance. After a brief outgrowth without antibiotics, interference efficiency is scored as the relative amount of antibiotic-resistant colonies. The results showed a strong correlation (r=0.89), indicating that CHAMP-derived binding affinities are also predictive of interference activity *in vivo* (Figure 10E). Moreover, our observations also help to rationalize the *T. fusca* self-avoidance mechanism. *T. fusca* encodes two Type I-E CRISPR loci. The first locus has a 5'-GGACCG PAM (ABA lower than our detection limit), whereas the second is 5'- GCTCAC PAM (ABA: ~1.9 k_BT). These are some of the lowest affinity PAMs and are predicted to strongly disfavor Cascade binding and R-loop formation. In sum, CHAMP profiling recapitulates DNA binding affinities measured via EMSAs *in vitro* and is highly correlated with *in vivo* interference activity.

Profiling off-target CRISPR-Cas DNA binding activity

To delineate the sequence determinants that influence Cascade-DNA interactions, I next analyzed the ABA for all DNA molecules with single or double substitutions along a 35-nt region that includes the first three positions of the PAM and the target DNA (Figure 11). CHAMP profiling yielded information for all possible single-base substitutions with an average 3,000-fold coverage (Figure 11A). As expected, substitutions in the PAM region reduced the ABA substantially, with the second position being most critical for Cascade

binding (Figure 11A). Prior structural and biochemical studies have established that every sixth nucleotide is unpaired and flipped out in the Type I-E Cascade-DNA complex[77,88–90]. A clear signature for these flipped-out base positions is also evident in the CHAMP profiling data (Figure 11A). A recent report identified that flipped out bases interact with a molecular relay of Cse2-encoded arginines[91]. Interestingly, our results indicate that substituting flipped-out bases with a thymidine mildly stabilized the Cascade complex. This interaction highlights additional Cascade-specified sequence preferences at the flipped out nucleotide positions.

I developed a simple model to better quantify how substitutions along the PAM and the target DNA affect Cascade binding (Figure 11B-D). This model considers a position-dependent penalty for all single base substitutions (Figure 11C) and a position-independent weight that accounts for the identities of each target and substituted base (Figure 11D). This model has fewer parameters than position weight matrices[92], but nonetheless described ~90% of the variance in the experimental data (Figure 11B). To further constrain this model, we acquired a second CHAMP dataset with a second crRNA-Cascade complex targeting a different DNA sequence. The model accurately described both independent CHAMP datasets acquired with two different crRNAs and corresponding DNA libraries ($r = 0.91$) (Figure 11B). Analysis of the base substitution penalties clearly highlights the importance of the PAM, as well as the PAM-proximal nucleotides (*i.e.* seed region) in modulating the affinity of Cascade for DNA. The overall substitution penalties decrease with increasing distance from the PAM (Figure 11C). This pattern has been recently observed for other CRISPR-Cas systems,[93], and likely reflects the initiation and directional formation of an R-loop starting from the seed region[82,84]. As expected from structural studies of the *E. coli* Cascade complex, the five flipped-out bases do not contribute to RNA-DNA interactions. Hence, substitutions at these sites may even be stabilizing, possibly due to DNA-protein interactions (Figure 11C). Overall, substitutions from any nucleotide to thymidine were mildly preferred, whereas substitutions from thymidine to any other

nucleotide were further destabilizing, though this should be considered in light of the preference for thymidines at flipped-out base positions (Figure 11A). I also analyzed the ABAs for all double nucleotide substitutions along the same 35-nt PAM and target DNA region (Figure 11E). The data highlights the importance of the PAM_{-2} position for controlling Cascade binding, as well as the easy tolerance of having any two flipped out base substitutions. In the seed region, single substitutions are already poorly tolerated and reduce ABAs significantly. Therefore, a second mismatch in the seed reduces the ABA to near-background levels, while a second mismatch in PAM-distal positions are often tolerated. Two substitutions in the PAM-distal sequence only marginally destabilized the Cascade-DNA complex.

Surprisingly, our data and model also reveal an additional periodicity in base-substitution penalties centered between the flipped-out bases (Figure 11C, Figure 11E). This periodicity results in an overall decrease in mismatch penalties every three nucleotides (e.g., at +3, +6, +9, etc.). A close inspection of the high-resolution *E. coli* Cascade structure reveals that every third base pair is puckered due to steric clashes between the RNA-DNA duplex and several residues in the Cas7 subunit (Figure 11F, Figure 11G). Six repeats of the Cas7 subunits polymerize along the crRNA to form the backbone of the Cascade complex. These subunits are likely to give rise to the three-nucleotide periodicity observed in our model and dinucleotide ABA data. Moreover, these residues are highly conserved amongst divergent Type I-E CRISPR-Cas systems (Figure 37), suggesting that they may play a role in Cascade assembly. Overall, our results highlight an unanticipated three-nucleotide periodicity in Cascade-DNA binding penalties that reduce the overall fidelity of RNA-DNA binding.

Cas3 recruitment requires perfect base pairing in the seed region

Complementary base pairing within an eight nucleotide PAM-proximal seed region is necessary for efficient interference *in vivo*[76,94–96]. However, CHAMP profiling revealed pervasive off-target DNA binding by Cascade (Figure 11). Therefore, we reasoned that subsequent binding of the Cas3 nuclease may constitute an additional sequence-dependent proofreading mechanism. We investigated this possibility with three-color CHAMP experiments that measured the degree of Cas3 recruitment to DNA-bound Cascade (Figure 13A). Fluorescent Cascade, Cas3, and alignment markers were spectrally separated into three distinct emission channels. After adding alignment markers, Cascade was introduced into the chips at a sufficiently high concentration to bind the majority of DNA clusters. Next, a saturating concentration of Cas3 was introduced into the same chip and CHAMP data was acquired (Figure 13B, see Appendix D). While most clusters showed a high degree of Cas3 recruitment, a large subset of the clusters showed lower than expected Cas3 fluorescence (Figure 13B, inset). As expected, we did not see any Cas3 binding to the DNA clusters when Cascade was omitted from the chip, or on clusters that did not bind Cascade. These results suggest that Cas3 is recruited to Cascade in a DNA sequence-dependent manner.

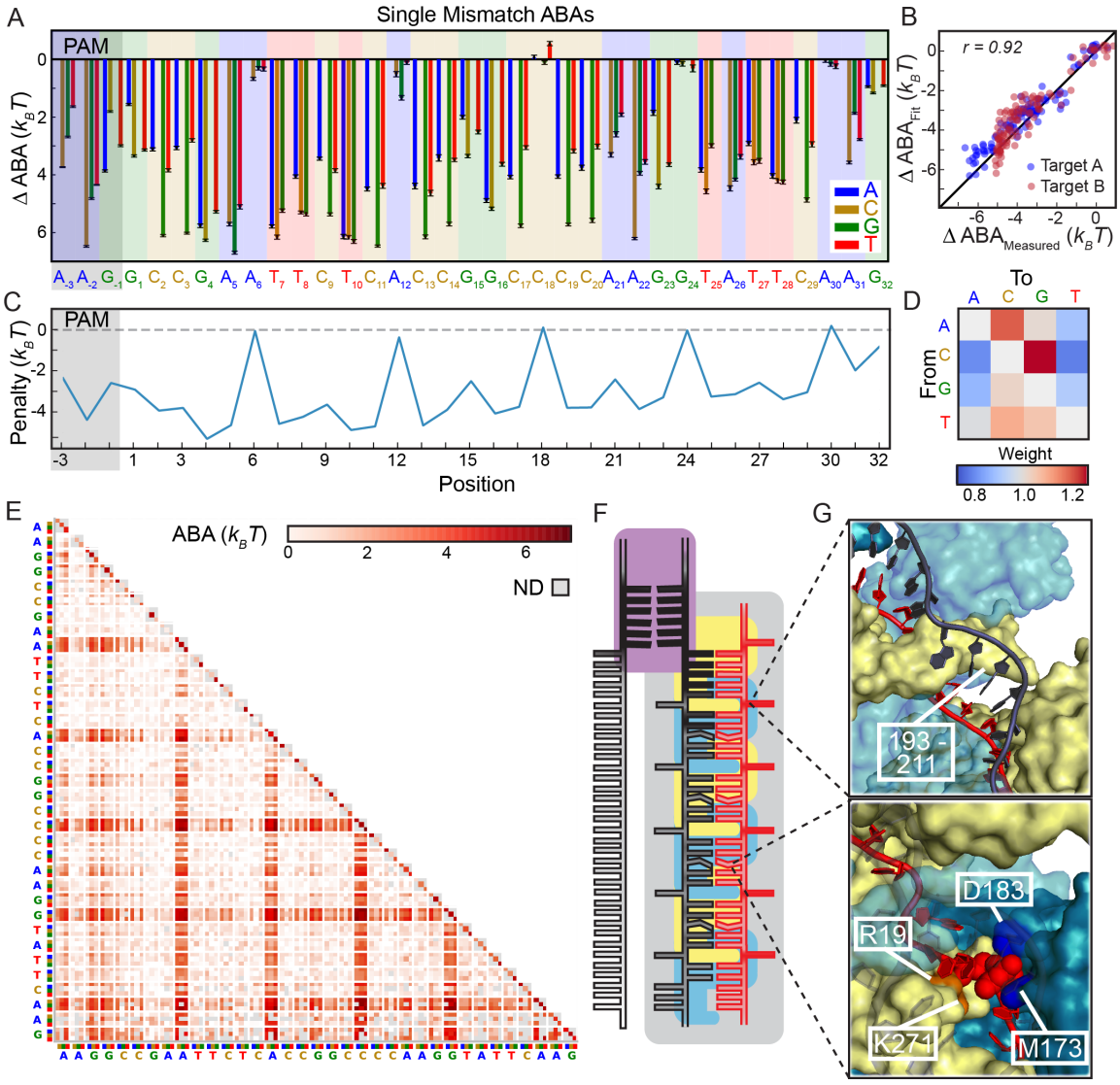


Figure 11. Comprehensive profiling of Cascade-DNA interactions.

Figure 11. Comprehensive profiling of Cascade-DNA interactions.

(A) The change in ABA for all 105 possible single-base substitutions along the minimal PAM and the target DNA. Negative values indicate a reduced ABA relative to the best PAM and perfectly paired DNA target. Error bars: S.D. obtained via bootstrapping. (B) CHAMP profiling was performed on two distinct DNA libraries (blue and red dots). The resulting data was used to construct a minimal binding model shown in (C) and (D) that accurately describes the data obtained from both CHAMP datasets. (C) Position-dependent substitution penalties and (D) position-independent nucleotide preferences obtained from the binding model. The model recapitulated the importance of the PAM, the PAM-proximal 'seed' region, as well as the occurrence of the flipped-out bases. A surprising three-nucleotide periodicity and a strong preference for thymidines at mismatched positions was also observed. (E) ABAs for all dinucleotide substitutions obtained with target A. The triangular matrix represents the average of CHAMP measurements acquired on two independent chips. The PAM is in the upper left-hand corner. Gray regions indicate insufficient data. (F) A schematic representation of *T. fusca* Cascade highlighting contribution of PAM positions -1 to -6, and the three-nucleotide periodicity. (G) Models representing the three nucleotide periodicity imposed by the protruding Cas7 finger (residues 193-211) (top) and steric clash with adjacent amino acids (R19, M173, D183 and K271) (bottom) based on *E. coli* Cascade[77].

We analyzed ~646,000 DNA clusters representing 10,810 unique DNA sequences to determine the requirements for efficient Cas3 recruitment. This dataset represented all extended PAM and single-nucleotide substitution variants, as well as 94% of double-nucleotide substitution variants along the target DNA sequence (Figure 9F). Approximately 450 DNA sequences showed a reduced ratio of Cas3 to Cascade fluorescent intensities relative to that of the fully complementary DNA target sequence. To better understand why Cas3 was not recruited at the same level to all DNA clusters, I focused on DNA sequences with single nucleotide substitutions along the PAM and the target DNA (Figure 13C). Comparing the Cas3 and Cascade fluorescent signals indicated that most DNA sequences fell on a diagonal line that indicates stoichiometric Cas3 recruitment, while those below the diagonal line indicate sub-stoichiometric Cas3 to Cascade ratios. The line of stoichiometric Cascade/Cas3 intensity was fit to all single-mismatch data with a mismatch in the fourth target position or greater. As expected, we did not observe any points above the diagonal (Figure 13C). Cas3 recruitment was partially compromised at nearly all non-AAG PAMs, as well as for target DNAs with a substitution in the first three PAM-proximal positions (Figure 13C). Using this information, I computed how sequence-dependent substitutions in the target DNA impact Cas3 recruitment. Their results are expressed as a Cas3 recruitment penalty calculated as the observed Cas3 average intensity minus the expected stoichiometric intensity given average Cascade intensity (Figure 13D). Surprisingly, our results revealed that mismatches in PAM_{-1} and +1 target positions strongly compromised Cas3 recruitment (Figure 13D). These data implicate the PAM, as well as the first few nucleotides in the seed region, as critical for Cas3 binding to a Cascade-DNA complex.

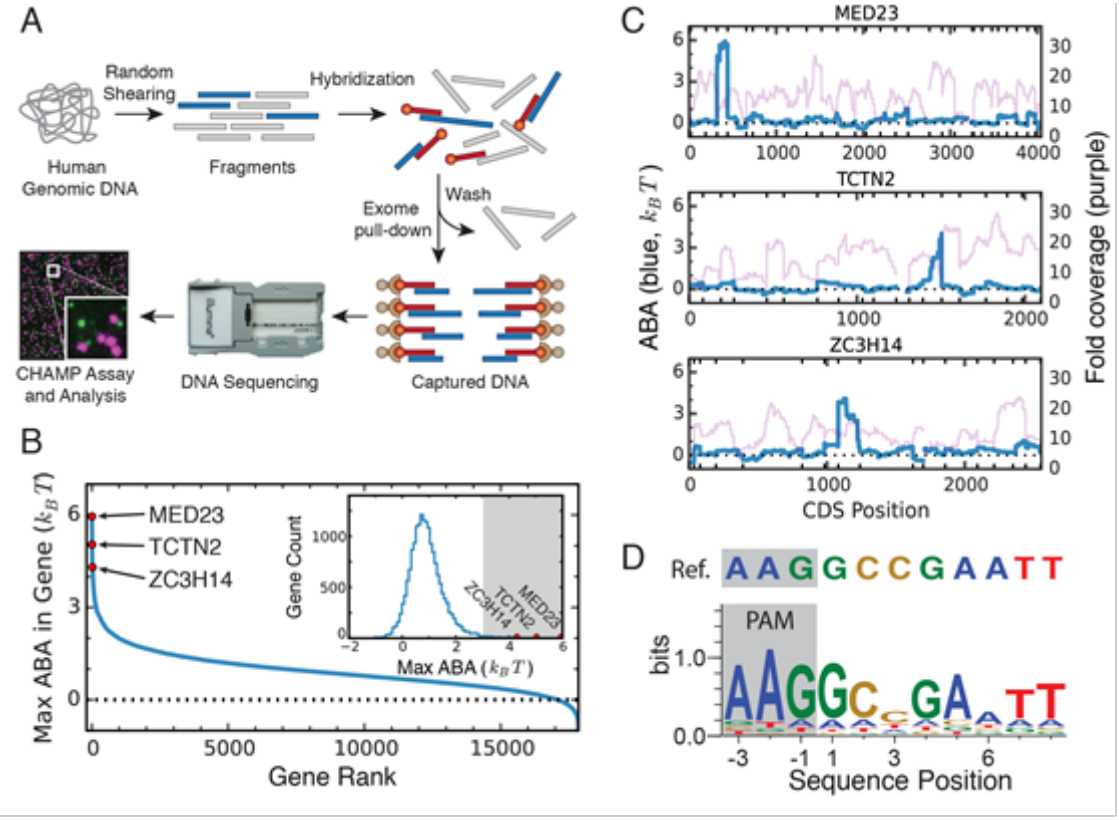


Figure 12. Profiling off-target Cascade binding in a human exome.

(A) The CHAMP-Exome analysis pipeline. Human genomic DNA is randomly sheared and enriched for exome sequences (blue) using standard oligonucleotide hybridization and bead pull-down protocols. After enrichment and adapter ligation, the exome is sequenced on a MiSeq chip, which is then used for CHAMP. Apparent Binding Affinities (ABAs) at each position in the exome were measured via CHAMP. (B) Maximum ABA values in each gene, ordered by rank. The dashed line indicates ABAs that fell outside of the experimentally defined cutoff for non-specific binding. Inset: histogram of genes that show measurable off-target binding. The gray zone indicates genes that had ABAs greater than 3 k_BT. Red dots in (B) indicate three representative genes with strong off-target binding sites, further described in (C). (C) Example high-affinity peaks. ABA is measured at each position in each gene using all reads overlapping that position. A high-affinity site thus appears as a peak in ABA whose width is a function of the DNA shearing length distribution. Shown are the measured ABAs at each position in a few genes containing high-ABA peaks. The ABAs spanning each gene are shown in blue (left y-axis) and the sequencing coverage in purple (right y-axis). Exon boundaries are shown as the minor ticks along the x-axis, and cause sharp changes in displayed ABA and coverage values. (D) Sequence logo generated from a 210-bp window centered around each of the ABA peaks > 3 k_BT. Image generated with WebLogo (Crooks et al., 2004).

Profiling off-target CRISPR-Cas binding in human genomic DNA

CHAMP uses a standard Illumina workflow and is immediately compatible with any nucleic acid library, including those derived from genomic preparations. We therefore extended CHAMP to profile CRISPR-Cas binding on human genomic DNA (Figure 12). To enrich for gene-coding regions, exome capture was used in conjunction with paired-end sequencing on an Illumina MiSeq sequencer (Figure 12A). The resulting sequenced MiSeq chip had an average 11-fold coverage for 17,862 human protein-coding regions from 7 million unique high-quality DNA clusters. This MiSeq chip was used to quantitatively assay off-target CRISPR-Cas binding. Remarkably, 37 genes showed at least one high-affinity CRISPR binding site (defined as ABAs > 4 k_BT) and ~200 genes showed moderate-affinity ABAs (> 3 k_BT). The precision of the off-target DNA sequence is defined by both the length distribution of the sheared exome fragments and the depth of coverage at each position (Figure 12B). Nonetheless, most genes harboring off-target sites showed a single, well-resolved ~200 bp-wide peak (Figure 12C).

The peaks with the highest ABAs represent genomic high-affinity off-target DNA binding sites. A subset of these peaks may also represent a combination of two lower affinity binding sites that are closer than our nominal resolution of 210 bp. Nonetheless, a logo analysis of all peaks with ABAs > 3 k_BT revealed a consensus sequence that matches closely with the expected critical determinants of off-target binding observed in our synthetic DNA libraries (Figure 12D). The consensus off-target site had a strong preference for an AAG PAM, with the second adenine giving the strongest signal (compare to Figure 10C). Second, off-target sites were highly enriched for the first eight basepairs of the target DNA sequence. One notable exception is the flipped-out base in the sixth position, which does not base pair with the crRNA (also see Figure 11). Consistent with binding data obtained from synthetic DNA arrays (Figure 11), mismatches are also tolerated at the third base, which has reduced basepairing with the crRNA. This data also highlights that an eight nucleotide PAM-proximal "seed" region is necessary for efficient binding, as has been

previously observed *in vitro* and via *in vivo* interference assays[76,94–96]. Here we demonstrate that CHAMP can profile off-target CRISPR-Cas binding sites in human genomic DNA, paving the way for rapid and quantitative profiling of off-target binding sites in patient-specific genomes.

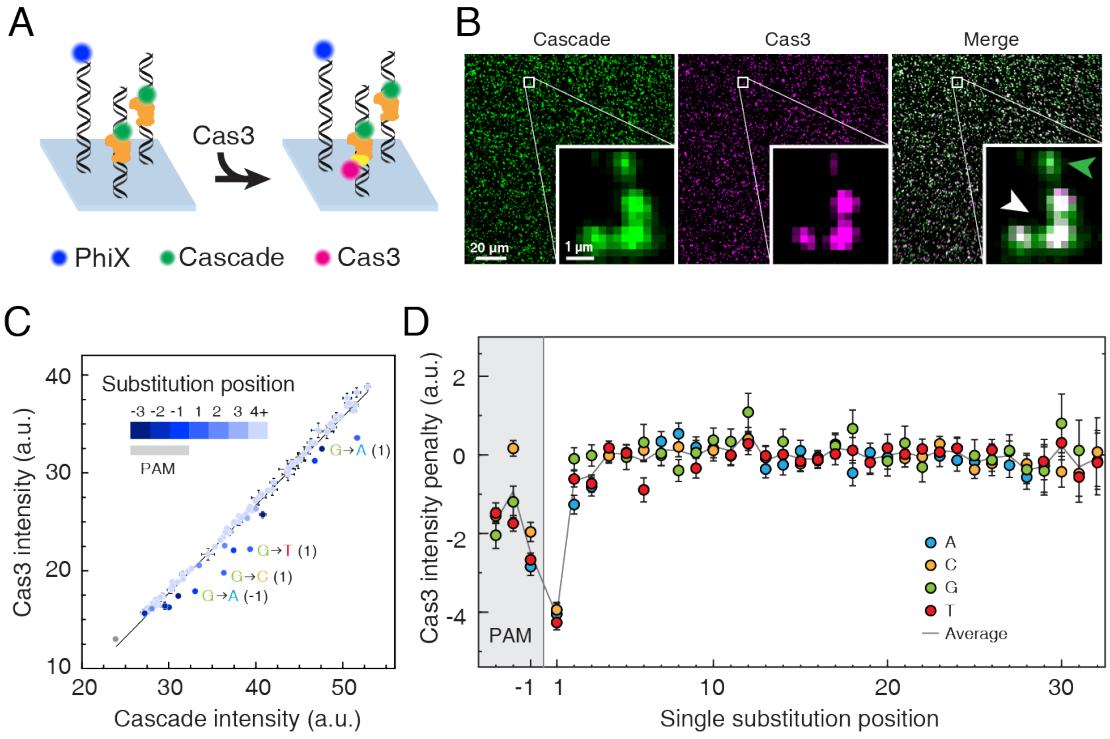


Figure 13. Three-color CHAMP reveals DNA sequence-dependent Cas3 recruitment.

(A) Experimental strategy overview. Fluorescent Cascade is first incubated in the regenerated chips. Next, fluorescent Cas3 is introduced into the same chip. (B) Most DNA-bound Cascade complexes readily bind Cas3 (white arrow, right inset). However, a small subset of clusters shows reduced Cas3 binding (green arrow, right insert). (C) Analysis of the fluorescent Cascade and Cas3 intensities at all sequences with a single nucleotide mismatch. Points below the diagonal indicate reduced Cas3 binding. Color bar indicates the position of the mismatch and the labels indicate the identity of the substituted bases. The gray point is a negative control indicating the background fluorescent intensity, as measured at non-specific DNA sequences on the same chip. Error bars: SEM of at least 213 independent clusters. (D) Analysis of the position-dependent Cas3 recruitment penalties. The solid line is an average of the three possible substitutions measured at each nucleotide position. Error bars: SEM.

Sequence-specific loss of Cse1 decreases the Cascade interference efficiency

We next used EMSAs and nuclease assays to further determine the mechanism of DNA-guided Cas3 recruitment (Figure 14). Cascade readily binds target DNA containing an $A_{-3}A_{-2}G_{-1}$ PAM. Surprisingly, the Cascade-DNA complex migrated as a faster mobility species when either this PAM was changed or when the +1 DNA position was mismatched relative to the crRNA (Figure 14A). Indeed, a DNA:RNA mismatch in the +1 position converted 80% of the Cascade complexes to the faster-migrating species. These effects were additive, as changing the PAM and the +1 position simultaneously resulted in nearly 100% of the faster-migrating sub-complex. Consistent with previous studies, we confirmed that this faster migrating species represents Cascade lacking the Cse1 subunit (Figure 36)[87,97]. Indeed, adding a large excess of free Cse1 could restore the mobility back to that of a complete Cascade complex (Figure 36). Cse1 physically interacts with Cas3 and loads the nuclease onto the target DNA[87]. Adding Cas3 resulted in a super-shift, but only when Cse1 was part of the Cascade complex (Figure 14A, Figure 14B). As expected, impaired Cas3 recruitment also reduced Cas3 nuclease activity when ATP and Co^{+2} were added to the reaction mixtures (Figure 14C and Figure 14D). Consistent with these *in vitro* studies, disrupting either the PAM or first few seed nucleotides also caused strong reduction in the plasmid-based *in vivo* interference assays (Figure 14E). These results reveal that DNA sequence-specific loss of Cse1 abrogates Cas3 recruitment and provides an additional proofreading mechanism for modulating CRISPR interference.

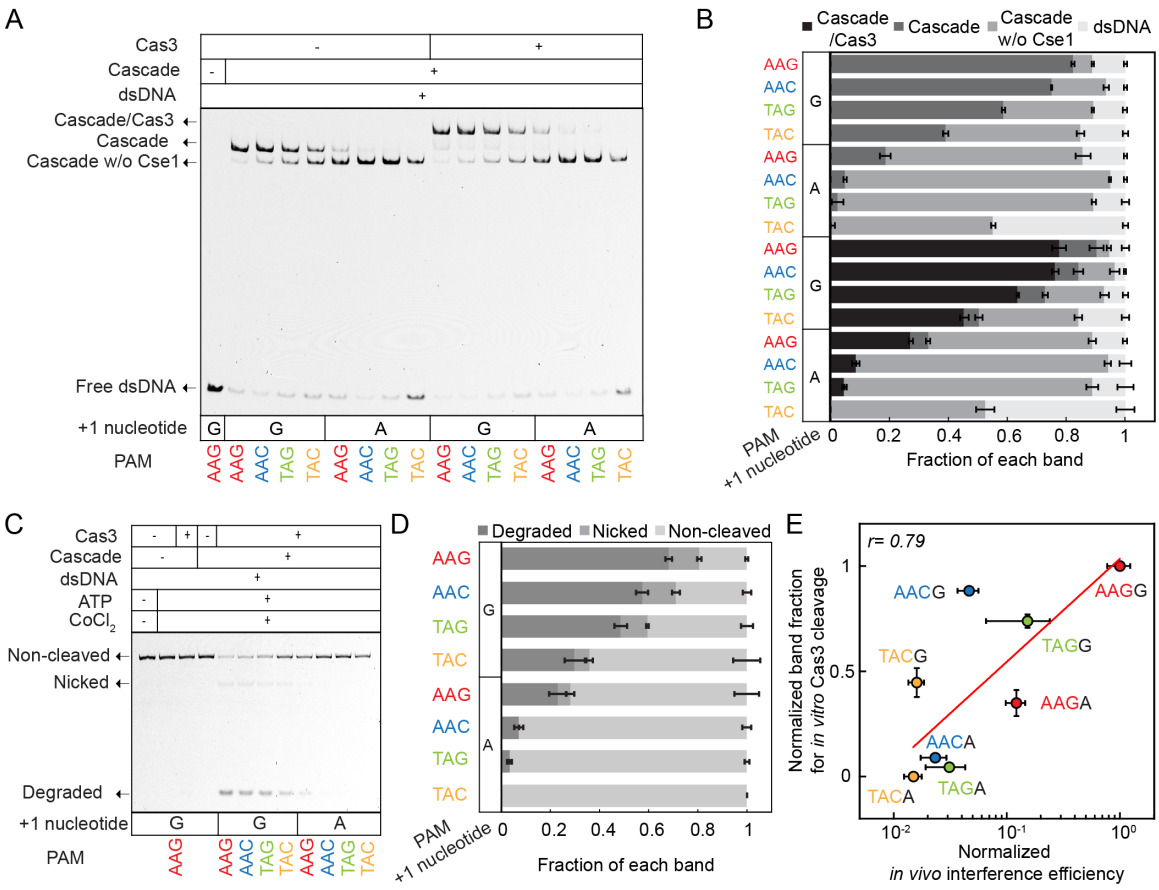


Figure 14. DNA-sequence dependent Cse1 dissociation provides an additional proofreading mechanism.

(A) Cse1 dissociation from the Cascade complex bound to DNAs with mismatches at the +1, -1, and -3 positions. Cas3 recruitment is Cse1-dependent, and is more impaired at mismatched sites containing these substitutions. Note that substitutions at the +1 position strongly promote Cse1 dissociation and abrogate Cas3 recruitment. DNA, Cascade, and Cas3 concentrations were 2 nM, 39 nM, and 1.1 μM, respectively. (B) Quantification of three replicates similar to (A). (C) Cas3 nuclease activity is strongly abrogated when mismatches are present in the +1 or PAM positions. Cas3 activity was Cascade, Co^{+2}, and ATP-dependent. DNA, Cascade, and Cas3 concentrations were 2 nM, 39 nM, and 650 nM, respectively. (D) Quantification of three replicates of (C). (E) *In vivo* interference is reduced when mismatches are present in the +1 or PAM positions. These results also agree with *in vitro* assays (r=0.79).

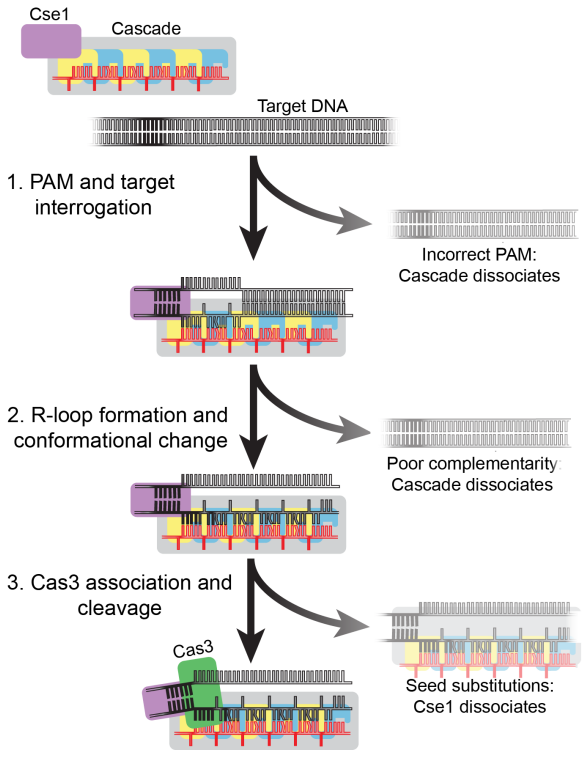


Figure 15. A DNA-sequence dependent proofreading mechanism by the Cascade/Cas3 effector complex.

Cascade first interrogates target DNA for an extended PAM sequence. Next, an R-loop is directionally extended from the PAM along the length of the crRNA. Cascade dissociation may be triggered by multiple mismatches between the RNA and DNA. An additional proofreading mechanism retains the Cse1 subunit only when the R-loop is properly formed within the 'seed' region. Finally, Cse1 recruits Cas3 for downstream CRISPR interference.

Cascade Binding and Interference Summary

Our findings reveal the biophysical parameters governing PAM recognition and DNA-binding at partially complementary target DNAs (Figure 15). *T. fusca* Cascade first identifies an extended PAM, possibly via hydrogen bonds with the PAM_{-4} nucleotide[77]. Further readout of the PAM_{-5} and PAM_{-6} positions may be mediated by indirect effects, such as changes in the major and minor groove widths at the PAM-proximal bases. These results are also broadly consistent with recent plasmid-based PAM-profiling experiments,

which also highlighted that diverse CRISPR-Cas systems—including the *E. coli* Type I-E Cascade—all decode an extended PAM[69].

Following PAM recognition and target DNA unwinding, an R-loop extends along the complementary target DNA. We utilized CHAMP to understand the effects of multiple sequence substitutions on Cascade-DNA interactions. In addition to identifying the importance of the PAM, "seed," and flipped-out bases, our analysis and modeling revealed an unanticipated three-nucleotide periodic interaction that reduced the relative penalty for having DNA-RNA mismatches at these positions. This periodicity likely arises due to a steric clash between basepairs in the R-loop and residues in each of the six Cas7 subunits. We speculate that these periodic contacts allow the crRNA to act as a scaffold during Cascade assembly. Indeed, a crRNA is required for assembly of the *E. coli* Cascade complex[90]. The crRNA is held in a conformation that maximizes interaction with the target DNA, possibly avoiding secondary structure formation by targets, as has been demonstrated in other RNA-guided nucleases[90,98–101]. The periodic mismatch tolerance also results in a much shorter 'effective' guide length (24-27 bp). Tolerating mismatches allows Cascade to recognize a wider array of targets, which is critical when defending against rapidly evolving phages.

Finally, by performing multi-color CHAMP imaging, we uncovered what appears to be a novel DNA-sequence dependent proofreading mechanism by the Cascade/Cas3 effector complex (Figure 15). Cas3 recruitment is dependent on the identity of the PAM, as well as perfect complementarity between crRNA and DNA in the +1 and +2 positions. These nucleotides interact with the Cse1 subunit of the Cascade complex. EMSAs and *in vitro* nuclease assays revealed that *T. fusca* Cse1 appears to dissociate from Cascade at intermediate PAMs or when there are mismatches between the crRNA and the first three nucleotides of the target DNA. The functional significance of this position was further confirmed with *in vivo* plasmid interference assays. The sensitivity of Cse1 retention and

subsequent Cas3 association increases the specificity of the overall system for the seed and PAM regions, recapitulating *in vivo* results (Figure 10E and Figure 14E).

COMPUTATIONAL METHODS

Aligning Fluorescent Images and FASTQ Points: Overview

To identify the DNA sequence of each cluster, I developed an image-processing pipeline to process images collected by our TIRF microscope (Figure 16A-B). To decode each cluster's sequence, its position was correlated to the corresponding record in the FASTQ file generated at the end of each MiSeq run. For each identified cluster, the FASTQ file reports the specifying lane, tile, and relative x-y coordinates, as shown in Figure 17. This FASTQ-supplied spatial information is reported in an arbitrary coordinate system that is scaled, rotated, and translated relative to our fluorescent images. An additional confounding factor is that FASTQ files do not report all fluorescent clusters (e.g., clusters that did not pass Illumina-specified quality control filters). In addition, some Illumina-reported clusters may also not light up in our fluorescent images. This may occur due to errors in the Illumina cluster identification pipeline, or possibly due to incomplete fluorescent labeling of the cluster during our experiments. As such, the mapping problem required finding the rotation, scale, x-offset, y-offset, and chip surface (both surfaces are imaged in a MiSeq chip) which best align the FASTQ points and imaged clusters. I accomplished this through two alignment stages: rough alignment and precision alignment, discussed below.

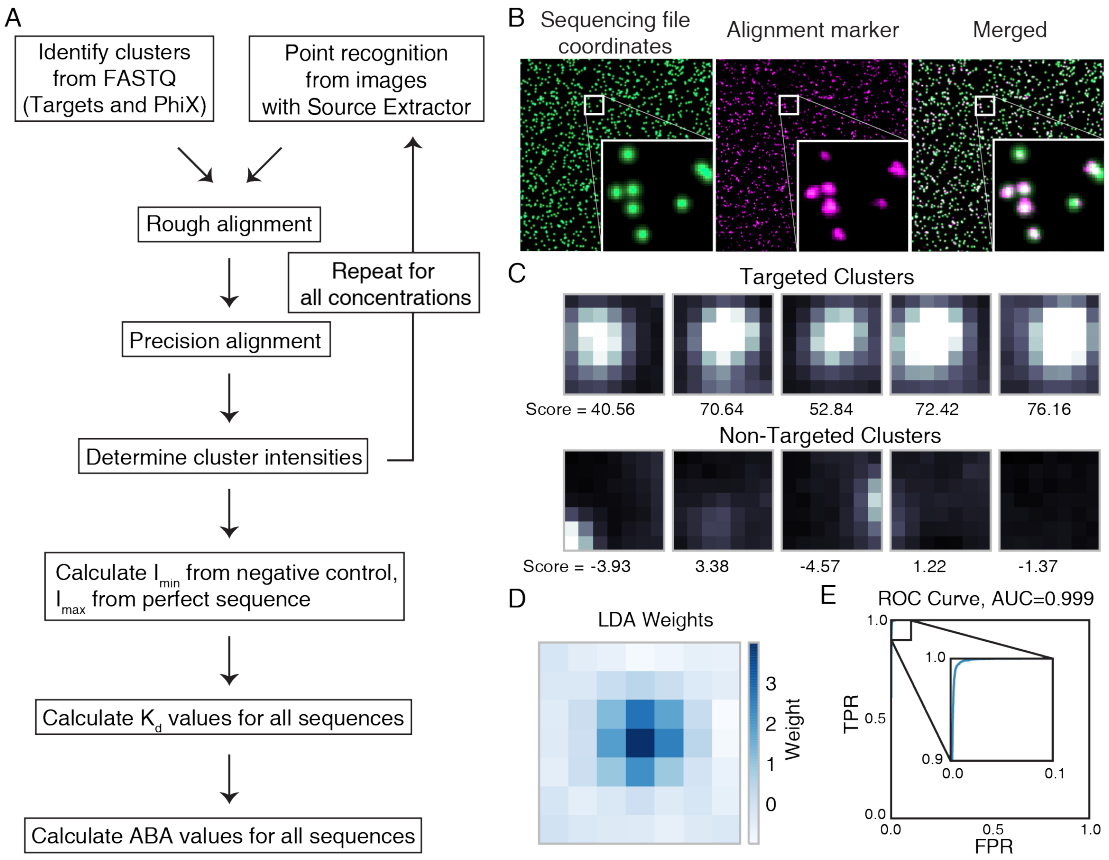


Figure 16. Cluster identification and linear discriminant analysis (LDA).

(A) Flow chart for cluster identification. (B) An example alignment. The first image shows a synthetic image representing the alignment marker coordinates, each represented by a symmetric Gaussian. These coordinates are found by mapping all reads against the PhiX genome, and aligning the mapped reads with the second image, a TIRF microscope image with fluorophores attached to all alignment markers. The third image shows the overlap of the synthetic and experimental images (overlap seen as white). (C) Example 7x7 pixel images centered on aligned FASTQ points for targeted and non-targeted clusters. (D) Linear discriminant analysis (LDA) was used to train pixel weights using sub-images as in (C) from sequences known to be on or off. Shown are the trained weights. 7x7 pixels sub-imaged were found to be optimal. To calculate intensity scores for K_d calculations, these weights, with negative values set to zero, are multiplied by the corresponding pixel values and summed. (E) The ROC (receiver operating characteristic) curve using LDA scores from (D) for classification of a test set of approximately 75,000 points. Perfect target A sequences were used as true positive rates (FPR), and non-target sequences as false positive rates (TPR). The extremely high area under the curve (AUC) of 0.999 indicates both very good alignment of the sequence coordinates and microscope images, as well as high fidelity of the chemistry in illuminating the correct clusters and only the correct clusters.

For the purposes of internal calibration, Illumina requires a percentage of each MiSeq run, typically 5-10% of all clusters, to be DNA from the small, thoroughly characterized phiX bacteriophage genome. Separate adapter chemistry is used for this phiX library, which can be accurately and specifically illuminated on any chip using complementary oligonucleotides. The phiX clusters do not contain a run-specific index barcode and are thus not demultiplexed as normal reads, but can be determined by mapping reads to the phiX genome. These phiX clusters provide a convenient resource for a variety of purposes, including alignment, categorization and intensity training, and as a control. We illuminated the phiX clusters by hybridizing them to a dye-conjugated oligo (Atto647-PCP or Cy3-PCP) during cluster re-generation and used the resulting fluorescent signals to align our fluorescent images with the corresponding FASTQ records.

Stage 1: Rough Alignment

The rough alignment was performed through cross-correlation of FASTQ points and images using fast Fourier methods[102]. Briefly, each FASTQ tile was converted to an image, with each cluster represented as a radially symmetric Gaussian with σ of 0.25 μm, a typical cluster size. Cross-correlation was then performed via the formula

$$Cross\ correlation = |\mathcal{F}^{-1}[(\mathcal{F}F)^* \cdot \mathcal{F}T]|$$

with zero-padding sufficient to accommodate any offset, where $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the fast forward and inverse 2D Fourier transforms, * is the complex conjugate, F is the FASTQ image, and T is the TIRF image. This allowed consideration of all x-y offsets (translation) in a computationally efficient manner, though did not inherently consider rotation or scale. The log-polar transform has been used in some applications to incorporate rotation and scale information into cross-correlation methods, but did not work well here. For each TIRF image, the maximum cross-correlation was first found against two FASTQ tiles known from their position to not overlap the TIRF image in order to measure background noise level, after which correlations above a signal-to-noise cutoff of choice, 1.4 in the

current work, indicated a good alignment. In order to achieve our first alignment, I first found good initial guesses for rotation and scale. I then exhaustively sampled a local grid around these estimates of rotation, scale, and parity to find the first alignment. With reasonable estimates for these parameters, the Fourier-based alignment can be performed within 45 seconds on a desktop computer.

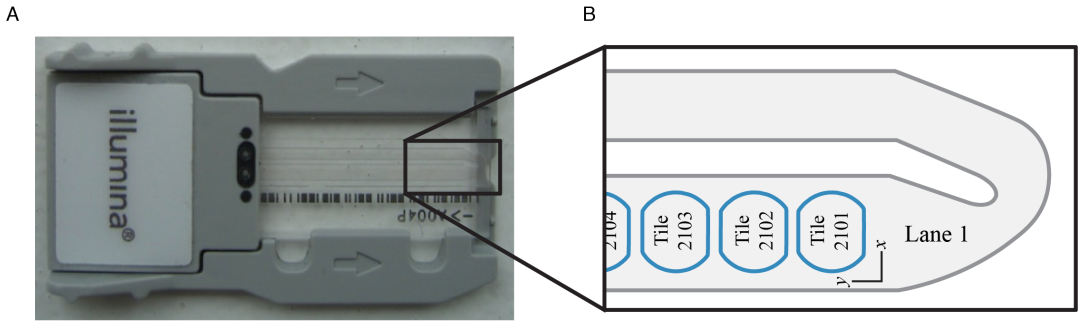


Figure 17. Illumina MiSeq Chip Coordinates.

(A) An Illumina MiSeq chip. (B) A schematic of the MiSeq microfluidics channel and v3 coordinate system. On Illumina machines, the location of each DNA cluster is specified by lane number, tile number within that lane, and x-y coordinates within that tile. A MiSeq chip has only one lane, Lane 1. Within the lane, tiles are numbered with four digits which indicate, in order, surface (both surfaces are imaged), swath (MiSeq has only one swath), and position along the swath, which in MiSeq v3 goes from 01 through 19. This comes to a total of 38 tiles per MiSeq v3 chip. Finally, x-y coordinates within each tile have a consistent, though arbitrary, footprint, with an x range of approximately (1700, 30000) and y range of approximately (1800, 25300). The orientation of the x and y axes is shown.

Stage 2: Precision Alignment

Following rough alignment in the alignment marker channel, I performed precision alignment via what I call constellation mapping in all channels. First, cluster location information was extracted from the TIRF images. I used the astronomy software Source Extractor to fit two-dimensional Gaussian functions to the fluorescent clusters[103]. Next, I found the nearest neighbors of FASTQ points in imaged cluster space and vice-versa using kd-trees[104]. Two points which were nearest neighbors of each other in both directions were termed a mutual hit. Due to accrued noise – missing data in FASTQ space, missing data in

imaged cluster space, and imperfect Gaussian calling – mutual hits were not by themselves high-confidence mappings. I further subcategorized mutual hits by the statuses of other nearby clusters. If cluster A and FASTQ point B were mutual hits and no other cluster X or FASTQ point Y consider A or B nearest neighbors, then the mutual hit was termed an exclusive hit. If there was another cluster X whose nearest neighbor was FASTQ point B, or another FASTQ point Y whose nearest neighbor was cluster A, then the status of hit AB was determined by the distance to the closest such X or Y. If the closest such X or Y was more than 1.25 microns away – the diameter of a typical cluster – AB was termed a good mutual hit; otherwise AB was called a bad mutual hit. Using exclusive hits and good mutual hits, we have approximately the same constellation of points in the two spaces. I then performed linear least squares fitting between these two constellations to determine the final alignment. The precision alignment process, including both constellation identification and least squares fitting, is typically performed within 2.5 seconds on a desktop computer.

Calculating Cluster Intensity

Machine-learned linear weighting of pixels was used to calculate the fluorescent intensity of each cluster. (see Figure 16C-E) For training, I used an experiment with only phiX clusters illuminated and restricted the analysis to exclusive and good mutual hits. Seven by seven pixel squares were extracted around each of these FASTQ points and linearized into feature vectors. Linear Discriminant Analysis (LDA) was then used to find pixel weights that best capture the intensity of a given cluster and penalize the intensity of neighboring clusters. The positive weights were used to calculate raw cluster intensities. To correct for variation in laser intensities across fields of view, cluster intensities were normalized within each run. The mode of pixel intensities of each image was calculated, and the intensity calculations in each image normalized by the mode of the given image divided by the median of all modes.

Calculating the apparent dissociation constant and binding affinity

Calculation of the apparent K_d value was performed for each sequence via curve fitting to the Hill equation (without cooperativity):

$$I_{obs} = \frac{I_{max} - I_{min}}{1 + \frac{K_d}{x}} + I_{min}$$

where I_{min} is the background intensity, I_{max} is the typical intensity of a fully saturated cluster, and the concentration values x and cluster intensity values I_{obs} are derived from the concentration gradient experiment. I_{min} is calculated as the median intensity of negative control clusters in the lowest concentration point. I_{max} is determined separately for each concentration to normalize small systematic errors between concentrations. The key observation is that due to very slow photobleaching rates, at higher concentrations where the perfect target sequence clusters have become saturated with Cascade, the perfect target sequence clusters can be used as a reference to normalize between concentrations. To this end, I_{max} is calculated in two steps, using only clusters of the perfect target sequence. First, the K_d and a temporary, constant I_{max} , call it $I_{max,const}$, are fit jointly on the perfect target sequence clusters using information from all concentrations. Second, for each concentrations where median I_{obs} is greater than 90% of the fit $I_{max,const}$, I_{max} is solved for from the above equation, using the observed median cluster intensity as I_{obs}. At all preceding concentrations, $I_{max,const}$ is used. These values of I_{min} and I_{max} are then used to fit K_d for all other sequences. Finally, given a K_d from a particular sequence and from a negative control sequence, call it $K_{d,NC}$, the apparent binding affinity (ABA) is given by

$$ABA = \ln\left(K_{d,NC}\right) - \ln\left(K_d\right),$$

chosen so that more positive indicates more binding. Error bars indicate the standard deviation of bootstrap K_d and ABA values. Figure 18 shows example average and 90% confidence errors for ABA as a function of number of clusters.

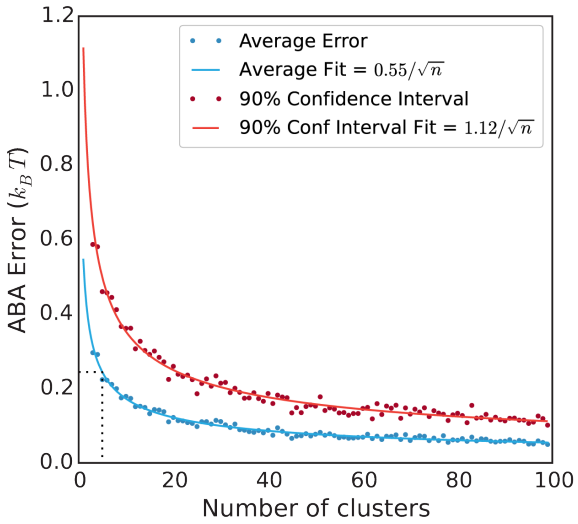


Figure 18. Estimating the error in the ABA.

Bootstrap ABA values were fit with all numbers of clusters between 3 and 100. Shown are the average errors compared with ABA (blue points), as determined for all clusters and 90% confidence intervals (red points). The gray dotted line shows our cutoff of 5 clusters, with average ABA error of approximately 0.2 k_BT. Solid lines indicate a fit to the data.

The source code for cluster identification, spatial registration, and binding affinity calculations is available via GitHub (https://github.com/finkelsteinlab/champ).

Position-Transition Model

In order to reduce dimensionality and aid insight regarding binding affinity changes from nucleotide substitutions in the PAM and target sequences, I developed what I call the position-transition model for change in apparent binding affinity (ΔABA), which can be written as:

$$\Delta\text{ABA} = \sum_{i=1}^{35} p_i \, t(r_i, s_i)$$

where p_i is the penalty, r_i is the reference base, and s_i is the sequenced base in the i^{th} position, and $t(x, y)$ is the position-independent transition weight from x to y. The summation is carried out over all 35 positions in the minimal three-nucleotide PAM and the protospacer.

For computational efficiency, I cast this in matrix form. I represented each sequence as a 35-by-12 indicator matrix S with rows representing each sequence position and columns representing each non-identity transition. The position penalties and transition weights were represented as vectors p and t. Then the above is written as

$$\Delta\,\text{ABA} = S : (p \otimes t)$$

where : is the Frobenius inner product and $\otimes$ is the outer product. This was linearized and concatenated into multiple-sequence sparse matrices in the natural way and fit using non-linear least squares. I removed model degeneracy by having multiple reference sequences and normalizing the transition vector to have mean value one.

DISCUSSION

CHAMP repurposes sequenced and discarded chips from modern next-generation Illumina sequencers for high-throughput association profiling of proteins to nucleic acids. A key difference between CHAMP and prior NGS-based approaches is that it does not require any hardware or software modifications to discontinued Illumina sequencers[63,65,64]. In CHAMP, all association-profiling experiments are carried out on sequenced MiSeq chips and imaged in a conventional TIRF microscope. CHAMP's computational strategy uses phiX clusters as alignment markers to align the spatial information obtained via Illumina sequencing with the fluorescent association profiling experiments. This strategy offers

three key advantages over previous approaches. First, using a conventional fluorescence microscope opens new experimental configurations, including multi-color co-localization and time-dependent kinetic experiments. The excitation and emission optics can also be readily adapted for FRET, and other advanced imaging modalities. Second, complete fluidic access to the chip allows addition of other protein components during a biochemical reaction. Third, the computational strategy for aligning sequencer outputs to fluorescent datasets is applicable to all modern Illumina sequencers, including the MiSeq, NextSeq, and HiSeq platforms. Indeed, we also used the CHAMP imaging and bioinformatics pipeline to regenerate, image, and spatially align the DNA clusters in a HiSeq flowcell, providing an avenue for massively parallel profiling of protein-nucleic acid interactions on both synthetic libraries and entire genomes. Future extensions will leverage on-chip transcription and translation (e.g., ribosome display) to facilitate high-throughput studies of RNA or peptide association landscapes. These studies will permit quantitative biophysical studies of diverse protein-nucleic acid interactions.

Cascade interrogates an extended PAM and recognizes mismatched DNA targets

Using CHAMP, we profiled the biophysical properties governing interactions between target DNA and the Type I-E CRISPR-Cas effector complex. Our findings reveal the biophysical parameters governing PAM recognition and DNA-binding at partially-complementary target DNAs. *T. fusca* Cascade first identifies an extended PAM, possibly via hydrogen bonds with the PAM_{-4} nucleotide as suggested by a recent high-resolution structure of the *E. coli* Cascade-DNA complex[77]. Further readout of the PAM_{-5} and PAM_{-6} positions may be mediated by indirect effects, such as changes in the major and minor groove widths at the PAM-proximal bases. These results are also broadly consistent with recent plasmid-based PAM-profiling experiments, which highlighted that diverse CRISPR-Cas systems—including the *E. coli* Type I-E Cascade—all decode an extended PAM[69].

Following PAM recognition and target DNA unwinding, an R-loop extends along the complementary target DNA. Using CHAMP, we probed the effects of multiple sequence substitutions on Cascade-DNA interactions. In addition to identifying the importance of the PAM, "seed," and flipped-out bases, our analysis and modeling revealed an unanticipated three-nucleotide periodic interaction that reduced the relative penalty for DNA-RNA mismatches at these positions. A re-analysis of previously reported *E. coli* Cascade plasmid interference assays also shows the same three-nucleotide periodicity[94]. Here, we propose that this is likely a general structural feature shared by other Type I-E systems and that it likely arises due to a steric clash between basepairs in the R-loop and residues in each of the six Cas7 subunits. The crRNA is required for assembly of the *E. coli* Cascade complex[90], and we speculate that these periodic contacts allow the crRNA to act as a scaffold during Cascade assembly. The crRNA is held in a conformation that maximizes interaction with the target DNA, possibly avoiding secondary structure formation by targets, as has been demonstrated in other RNA-guided nucleases[90,99,101]. This periodic mismatch tolerance was also confirmed at off-target sites mapped to the human exome, further highlighting the importance of quantitatively mapping the influence of mismatches on CRISPR-DNA interactions with both synthetic and genomic DNA substrates.

A DNA sequence-dependent mechanism underlies Cse1 loss and CRISPR interference

By performing multi-color CHAMP imaging, we uncovered that Cas3 recruitment is dependent on the identity of the PAM, as well as perfect complementarity between crRNA and DNA in the +1 to +3 positions (Figure 13). These nucleotides interact with the Cse1 subunit of the Cascade complex. EMSAs and *in vitro* nuclease assays revealed that *T. fusca* Cse1 dissociates from Cascade at intermediate PAMs or when there are mismatches between the crRNA and the first three nucleotides of the target DNA. The functional significance of this position was further confirmed with *in vivo* plasmid interference assays

and also recapitulates previously published *in vivo* interference results with the *E. coli* Cascade complex[94].

In addition to identifying foreign DNAs, Cascade and Cas3 also promote primed spacer acquisition, where additional spacers are rapidly acquired from foreign DNAs that already contain a spacer in the CRISPR locus. Spacer acquisition requires the Cas1-Cas2 protein complex, which binds protospacer DNA and uses its integrase activity to insert the protospacer within the CRISPR array. Cascade can promote target acquisition at both perfectly matched spacers and mismatch-containing spacers that do not elicit strong interference[105–108]. Conformational control of the Cse1 subunit is emerging as a key paradigm for recruiting Cas1-Cas2 and redirecting the Cascade-Cas3 complex towards primed acquisition[108]. Here, we speculate that Cse1 undergoes a DNA-sequence dependent conformational change that renders it labile in the absence of Cas1-Cas2 complex. Future CHAMP studies with fluorescent Cas1-Cas2 and FRET-reporters of Cse1 conformational state will shed light on the mechanisms and sequence requirements for primed spacer acquisition.

Leveraging CHAMP for mapping protein-nucleic acid interactions on human genomes

Because CHAMP uses the standard Illumina workflow, it is immediately compatible with any nucleic acid library, including synthetic DNA, RNA, or genomic preparations. However, mapping CRISPR-DNA interactions on sequenced genomes presents additional computational challenges due to the random shearing lengths and uneven sequencing coverage. To address this challenge, we developed a bioinformatics pipeline that successfully identified off-target binding sites within a human exome with a ~200 bp effective resolution at an average 11-fold coverage depth. Higher resolution mapping can be readily achieved by shorter DNA fragments and greater sequencing coverage. Thus, CHAMP can be used to probe off-target CRISPR-Cas binding in any genome prior to

performing genome-editing. Further extensions will allow direct observation of both binding and cleavage at these off-target sites. As CRISPR-Cas systems continue to be developed for human gene modification, CHAMP and similar methods may become useful tools for rapidly and quantitatively assaying target specificity on individual patient's genomes.

Chapter 3: A meta-analysis of bat genomes and transcriptomes‡

INTRODUCTION

The bat order Chiroptera is one of the most common and diversely adapted orders of organisms on Earth. Bats form a disproportionately large portion of the number of mammal species, representing 925 of approximately 4,600 of all known mammal species, about 20%[109]. Several characteristics of bats make them inherently interesting, most uniquely including their ability to fly and echolocate. Bats have also gained notoriety for being important reservoirs of several deadly zoonotic viruses. They are established or conjectured viral reservoirs for the SARS coronavirus, Nipah virus, Hendra virus, and the Ebola virus[109,110].

Unfortunately, the diversity that makes bats interesting also makes them more difficult to study. Compare the flying fox, *Pteropus vampyrus*, a fruit eating Southeast Asian giant with a wingspan up to 1.5 meters, with the vampire bat, *Desmodus rotundus*, a South American bat which feeds primarily on blood, has the ability to run over ground, and weights just 25-40 grams[111–113]. In genetic studies, genes are often isolated using primers designed from closely related species, and this amount of divergence within the order Chiroptera makes isolating genes of interest from novel bat species very tedious and expensive. At the same time, the large divergence, paired with the relatively sparse data often used, has led to difficulty establishing the topology of the phylogenetic tree.

---

‡ This chapter draws on material from Hawkins JA, Kaczmarek ME, Press WH, Sawyer SL. A meta-analysis of bat genomes and transcriptomes. (In preparation). J.A.H., M.E.K., W.H.P., and S.L.S. designed the research. J.A.H. performed all computational methods and analyses. M.E.K. prepared the RNA for sequencing. J.A.H., W.H.P., and S.L.S. wrote the paper. All authors commented on the manuscript.

There are two principle phylogenic analyses with bats across the entire order Chiroptera. In 2002, Jones *et al*. combined 105 previous phylogenetic studies of Chiropteran families and subfamilies from publications as far back as 1970, placing 925 bats into one tree using supertree parsimony methods[114]. In 2011, Agnarrson *et al*. built a phylogeny *de novo* using molecular Bayesian methods on a single gene, cytochrome b (CytB), from 648 species[115]. These trees agree on much of the history of Chiroptera, but due to the host of different methods represented and the relative sparsity of the data, they unsurprisingly differ on a number of points with regard to the basic backbone of the tree. Since the time of these two analyses, a number of bats across the order Chiroptera have benefitted from the explosion of data that is next generation sequencing, suggesting the time is ripe to revisit the backbone of the Chiropteran phylogeny through use of the sequence information from all available genes.

The same data we wish to use to find the phylogenetic tree—namely, multiple sequence alignments (MSAs) of all available genes—is also highly valuable for guiding research into mechanisms of viral antagonism. To infect a cell, a virus must hijack healthy cellular proteins in order to enter the cell and replicate within it. Through natural selection, any host which has proteins too easily tricked by a given virus will tend to be replaced by hosts with mutations resistant to the virus. The virus, then, is under selective pressure to find a mutation in its own genome allowing it to hijack the new protein variant. And back and forth it goes through evolutionary time in what has become known as the host-virus arms race, with the amino acids at the interface in a state of constant flux[116,117]. This constant selective pressure, known as positive selection, leaves its imprint on the DNA sequences of the host species by producing an unusually high ratio of mutations which change the protein sequence to those that do not, known as dN/dS[118,119]. We can use multiple sequence alignments to measure this ratio and look for signatures of positive selection. Identifying Chiropteran genes under positive selection is of particular interest to human health due to the several bat viruses which are deadly for humans[120,121].

To address the above problems with the new wealth of sequencing data, we set out to perform a meta-analysis of available Chiropteran annotated genomes and transcriptomes, with three principal goals: a curated set of orthologous gene families, a high-confidence phylogeny, and positive selection measurements for each family of orthologous genes.

To this analysis, I, in collaboration with Maryska Kaczmarek in Dr. Sara Sawyer's lab, have also added new transcriptomic data and annotation for two African bats of interest, associated with the Ebola virus and its relative the Marburg virus: *Hypsignathus monstrosus* and *Rousettus aegyptiacus*. *H. monstrosus*, the hammer-headed bat, is known for its likely connection to the Ebola virus. Proving that a given species of any organism is the reservoir for Ebola has proven elusive, but many believe that bats are the reservoir since Leroy *et al.*'s 2005 paper in which they performed a broad test of more than a thousand small vertebrates near sites of recent Ebola outbreaks[122]. Live virus was not extracted in large quantity from any of the organisms, but several organisms were found to have immunoglobin G (IgG) specific to the Ebola virus. Chief among them were the bats, and chief among the bats was *H. monstrosus*. *R. aegyptiacus*, meanwhile, is an established reservoir for the Marburg virus[123,124].

Species	Genome / Transcriptome	Annotated Genes	N50	%GC	Assembly Count
Artibeus jamaicensis	Transcriptome	10,071	2,166	53.4	16
Carollia brevicauda	Transcriptome	3,954	1,284	51.3	12
Cynopterus sphinx	Transcriptome	6,232	1,653	49.8	12
Desmodus rotundus	Transcriptome	9,019	2,115	52.8	18
Eptesicus fuscus	Genome	13,248	2,235	54.2	n/a
Hypsignathus monstrosus	Transcriptome	7,875	2,040	49.8	17
Macrotus californicus	Transcriptome	4,375	1,557	51.9	12
Miniopterus schreibersii	Transcriptome	11,089	2,202	53.4	19
Murina leucogaster	Transcriptome	9,267	2,055	53.6	14
Myotis brandtii	Genome	12,674	2,229	53.1	n/a
Myotis davidii	Genome	12,353	2,223	53.2	n/a
Myotis lucifugus	Genome	12,386	2,214	53.2	n/a
Myotis ricketti	Transcriptome	4,868	1,401	51.1	12
Pteropus alecto	Genome	13,295	2,235	52.4	n/a
Pteropus vampyrus	Genome	13,145	2,232	52.3	n/a
Rhinolophus ferrumequinum	Transcriptome	6,764	1,761	53.8	12
Rousettus aegyptiacus	Transcriptome	9,714	2,235	52.8	18
Tadarida brasiliensis	Transcriptome	6,128	1,869	51.4	12
Homo sapien	Genome	13,206	2,301	52.2	n/a
Sorex araneus	Genome	12,190	2,280	55.7	n/a

Table 1. Chiroptera data overview.

Each species analyzed in this study, along with basic information concerning their data, including data type (genomic or transcriptomic), the number of genes here placed in orthologous gene sets, N50 and GC content of said genes, and, for bats with transcriptomic data, the number of assemblies constructed and analyzed. Transcriptomic data were all assembled and annotated as part of this study, while genomic data and annotations were all downloaded from RefSeq.

RESULTS

Data collection and assembly

The bat species analyzed in this study and the type of data associated with each are shown in Table 1. For each bat with annotated genomes, we downloaded the relevant RefSeq database from the National Center for Biotechnology Information (NCBI) website and extracted the protein and coding sequence of the longest isoform of each gene. More consistent isoforms were found after orthology search (see Methods). Genome assembly accession numbers, as well as basic assembly statistics, for each genome are given in Table 3. Human and common shrew, *Sorex araneus*, genomes were also included for use as outgroups. We collected RNA-seq data for *H. monstrosus* and *R. aegyptiacus* ourselves, sequenced on Illumina HiSeq machines (see Methods). For all other bats with available transcriptome data, we downloaded the raw sequencing reads from the Short Read Archive (SRA)[125–136].

For the sake of consistency, we used only transcriptome assemblies constructed using our own pipeline, even in the rare cases where authors made assemblies publicly available. Briefly, our pipeline consisted of removing adapter sequences with Trimmomatic[137], followed by using two of the most popular transcriptome assemblers, the De Bruijn graph-based Trinity[138] and TransAbyss[139] assemblers, with a range of input parameters. This resulted in multiple tentative assemblies per bat (see Table 1, Methods)[137–140]. The best-assembled contig for each gene among these assemblies was selected in the orthologous gene finding stage.

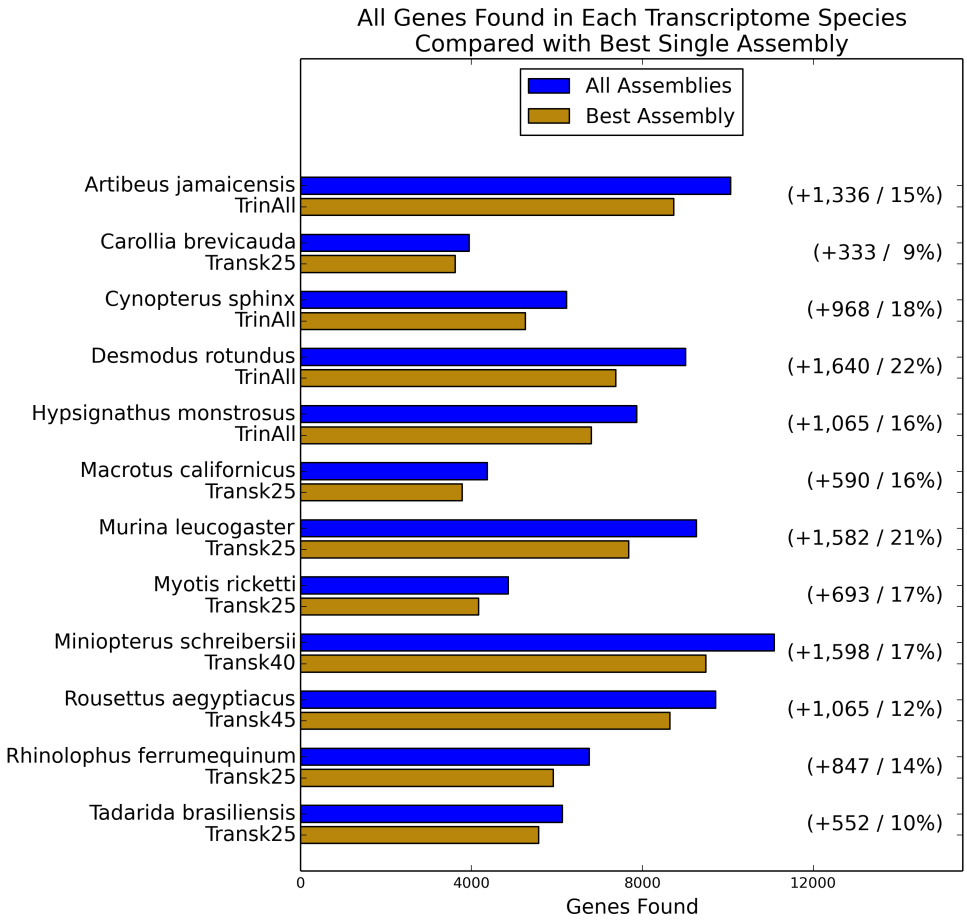


Figure 19. Use of multiple assembly methods improves recovered gene counts.

For each species, the number of genes placed in an orthologous gene family based on all assemblies is reported (blue), as well as the number of genes which could have been found from the best single assembly alone, had it been known a priori (yellow). Which assembly was best for each species is shown as the labels for yellow bars, where TrinAll indicates the Trinity assembly using all reads and TranskXX indicates the Trans-Abyss assembly using the De Bruijn graph of k-mers of length XX. Number of genes added through use of multiple assemblies over the best assembly and percentage increase are shown right of the bars. Use of multiple assemblies added 9-22% more annotated orthologs per species relative to the best single assembly.

Orthologous Gene Families

The search for orthologous genes was performed in two primary steps. First, we searched for orthologs in the genomic data sets. With genomic data, one is able to use syntenic information to predict orthology rather than paralogy. We used all-v-all Blast reciprocal best hits (RBH) of the protein sequences, filtered using three sources of syntenic information: public orthology predictions from BioMart, proximity via whole genome alignment, and proximity of similar neighboring genes[141–143]. Second, we searched for orthologs in transcriptomic data sets. We selected the best Blast RBHs of transcriptomic data against genes found in the genomic orthologous gene sets, filtered by search using HMMER[144], a hidden Markov model based homology search software package, and filtered by match length (see Methods).

My final results place 192,686 transcripts into 12,611 orthologous gene sets, of which 1,334 contain genes from all species and outgroups. Figure 19 shows the number of transcriptomic genes found by species. Also shown in Figure 19 is the number of genes we would have found in each species had we known a priori which assembly would perform the best for each bat and only assembled that one. With the additional work of analyzing multiple assemblies per bat, we were able to identify 9-22% more genes per bat relative to the best single assembly, representing thousands of added transcripts and improvements to the completeness of the gene network.

Multiple Sequence Alignment Cleaning

Manual inspection of many multiple sequence alignments (MSAs) of orthologous genes revealed a non-random source of error: the species were biased toward segregating by data type (Figure 19a). I.e., genomic data and transcriptomic data would tend to agree within data type but disagree between data types. Furthermore, the splits were observed to happen at sharp boundaries highly suggestive of exon boundaries. This effect is naturally explained

by the fact that we chose the longest isoforms for genomic species, even though the longest isoforms might not be expressed at high enough levels to appear in the transcriptomic data sets. Any systematic artifacts in transcriptomic or genomic data assembly and annotation would also contribute to this effect.

To ameliorate this bias, we developed a two-step cleaning algorithm for the MSAs (Figure 19b). First, we revisited each genomic gene and replaced it, if necessary, with the isoform closest to the consensus sequence. This resulted in improvements to 3,444 transcripts, with transcripts improving their match to the consensus sequence by an average of 8%, though there was a wide range of percent improvements (Figure 19c). Second, we removed exons if the species did not all agree on the exon structure. Specifically, we removed exons if all species did not agree on the aligned exon boundaries or if exon sequences differed in length by more than 10%. This cutoff was chosen because we observed it to be a transition point to high-gap exons in our data (Figure 38). Note that we intentionally did not use sequence identity or agreement to select for good or bad exons. One of the analyses we wish to perform is positive selection analysis, which measures the ratio between non-synonymous and synonymous mutations. Filtering exons based on whether the sequences matched would directly bias the data in favor of synonymous mutations.

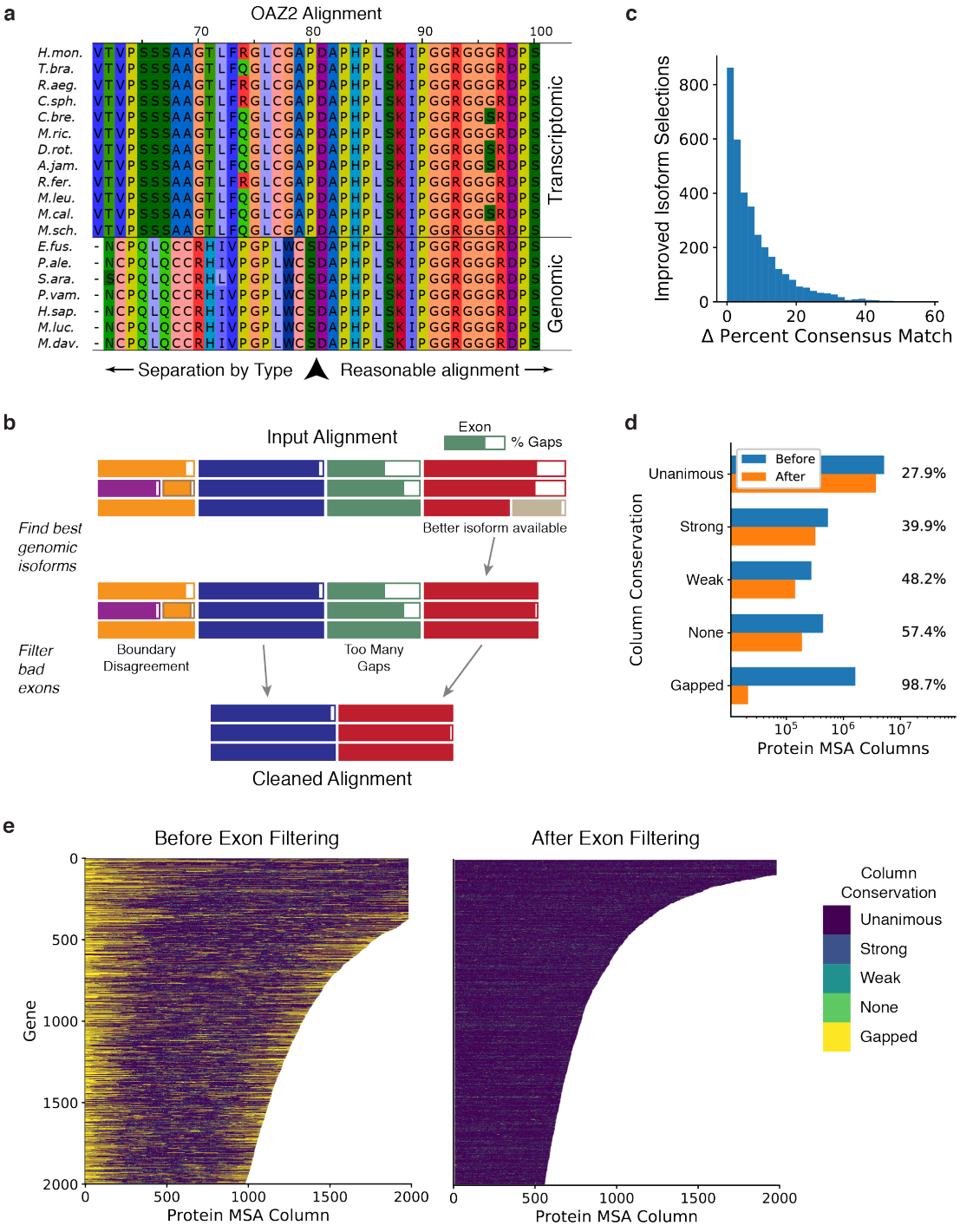


Figure 20. Multiple sequence alignment cleaning.

Figure 20. Multiple sequence alignment cleaning.

(a) During manual inspection, multiple MSAs were observed to have isoform selection biased toward separation of genomic and transcriptomic data, consistent with the observation that longest isoforms, used for genomic data, need not be expressed at high enough levels to be present in transcriptomic data. This produces non-random, non-gapped errors segregating by the artifact of data type. (b) To clean the alignments, we first revisit each genomic gene to choose the isoform which best matches the consensus alignment sequence. We then filter exons, where exons with disagreement about boundary positions in the alignment and exons with too many gaps are filtered out. (c) 3,444 genomic isoform selections were improved in the first cleaning step, shown as a function of the improvement in percent matching the consensus sequence, i.e. (percent matching after) – (percent matching before). (d) Our exon filtering strategy enriches for conserved sequences, filtering more weakly conserved sites at higher rates, even though the filtering strategy intentionally does not consider sequence composition. “Strong”, “Weak”, and “None” conservation categories are as defined by Clustal. Percentages show percent reduction in MSA column counts. (e) Column conservation of the first 2,000 columns of the longest 2,000 alignments before and after exon filtering.

To verify that our exon-filtering strategy improved the quality of the alignments, we checked that the filtered sequences have improved overall sequence conservation. Our strategy, as hoped, preferentially discriminates against more weakly conserved sites, filtering nearly 60% of non-conserved sites vs. only 28% for unanimous sites (Figure 19d). Gapped sites, being directly relevant to the filtering process, are nearly 99% removed. Figure 19e shows the positions and distribution of conserved sites in the first 2,000 MSAs. Many of the bad exons are at the ends of the alignments. There are a few reasons to expect this. The main reason is that the ortholog finding process scores sequences based on length of agreement, which naturally tends to include matching sections in the center. The ends are then more free to vary, with variability expected due to differences in isoform selection, as well as due to incomplete or incorrect transcript assembly. Transcripts tend to have less coverage near the ends, resulting in worse assembly. Incomplete assembly at the ends also helps explain why so many unanimous sites end up being filtered: correctly matched exons will still be filtered if partial assembly results in exons of different lengths. From these results, as well as manual inspection, the alignments have significantly fewer erroneous alignment columns after exon filtering.

Phylogenetic Analysis

I constructed the phylogenetic tree of our considered species using multiple strategies and software packages. First, using the 1,334 genes found in all species, we constructed the species tree with a partitioned nucleotide analysis—one partition per gene—in Mr. Bayes[13]. Next, using the same genes, we constructed the species tree with concatenated data using Mr. Bayes and RAxML[14]. Finally, we constructed the 1,334 gene trees with all species using Mr. Bayes and determined the species tree via quartet parsimony as implemented in ASTRAL[15]. Gene trees were computed with both nucleotide and amino acid sequence.

The final tree is shown in Figure 21a, with reported posterior probabilities given from the Mr. Bayes partitioned analysis. We refer to the final tree instead of a specific version of the final tree due to the strong consensus between methods. Methods used include Mr. Bayes with a partitioned model sampling over gamma model space; Mr. Bayes with concatenated data also sampling over model space; RAxML with concatenated data; and quartet parsimony of gene trees via the ASTRAL software package on several inputs, including nucleotide CDSs, amino acid sequence, and restrictions to the three codon positions. A summary of how the methods agreed or disagreed is shown in Figure 21b. All methods converged on nearly the same species tree. In fact, due to the large amount of data, all nodes resolved with 100% reported posterior probability in both Mr. Bayes analyses. The only species not consistent in every analysis were *C. sphinx* and *M. leucogaster*.

Also included in Figure 21b are comparisons with trees of Agnarsson, et al and Jones, et al[115,145]. All species placements in our final tree agree with these trees, with the exception of two previously controversial species, *R. ferrumequinum* and *M. schreibersii*; one particularly close node, *C. sphinx*; and one surprising placement, *M. leucogaster*.

The placement of *R. ferrumequinum* addresses the first branching of the order Chiroptera. The traditional division of order Chiroptera into Mega- and Microchiroptera, the large and small bats respectively, has been challenged in recent years as molecular phylogenetic analyses have gained prominence. An alternative history has been proposed, dividing bats into two suborders named Yinpterochiroptera and Yangochiroptera[146]. The microbat families Rhinopomatidae, Rhinolophidae, Hipposideridae and Megadermatidae are joined with the megabats to form the new clade Yinpterochiroptera, while the rest of the microbats form Yangochiroptera. This restructuring has gained recent support. See [147–149]. *R. ferrumequinum,* a member of family Rhinolophidae, is the only bat in our data which falls into this contested group. Our results side firmly with the division into Yinptero- and

Yangochiroptera, with the placement of *R. ferrumequinum* with the megabats in Yinpterochiroptera, the clade subtended by node B in Figure 21a.

The placement of *M. schreibersii* has also been unclear. Agnarsson's cytochrome B based phylogeny places the *M. schreibersii* just outside node H on our phylogeny. On the other hand, our placement of *M. schreibersii* agrees with the Jones phylogeny, as well as Hoofer et al., who argued that due to this placement and the large divergence, Miniopteridae deserved to be its own family[150].

The most surprising placement in our tree is that of *M. leucogaster*. In all of our trees, with 100% reported posterior probability where calculated, *M. leucogaster* disrupts the monophyly of genus Myotis. One must keep in mind, however, that the quantity of data here considered is so large it will tend to produce results with 100% probability at each node, pushing the question of credibility further upstream to the quality of the data. No data assembly and cleaning strategy, including our own, is perfect, and it is possible that sufficient errors remain in our data as to result in an erroneous topology of such closely related species. It at first even seems suggestive, given our previous observations of bias by data type, that the two Myotis species placed with *M. leucogaster*—*M. davidii* and *M. ricketti*—are both also transcriptomic while the other two Myotis species are genomic. However, if data type were the explanation for this placement, we would expect *M. leucogaster* to be the outgroup to *M. davidii* and *M. ricketti*, not between them. Furthermore, the relationships between the ranges of the bats in this clade roughly match our consensus phylogeny: *M. ricketti* lives in southeast Asia, *M. leucogaster* lives in southeast Asia and central China, *M. davidii* lives in central China, *M. brandtii* lives across Europe and parts of northern Asia, and *M. lucifugus* lives in North America[151–155]. These results suggest the classification of *M. leucogaster* may merit reconsideration.

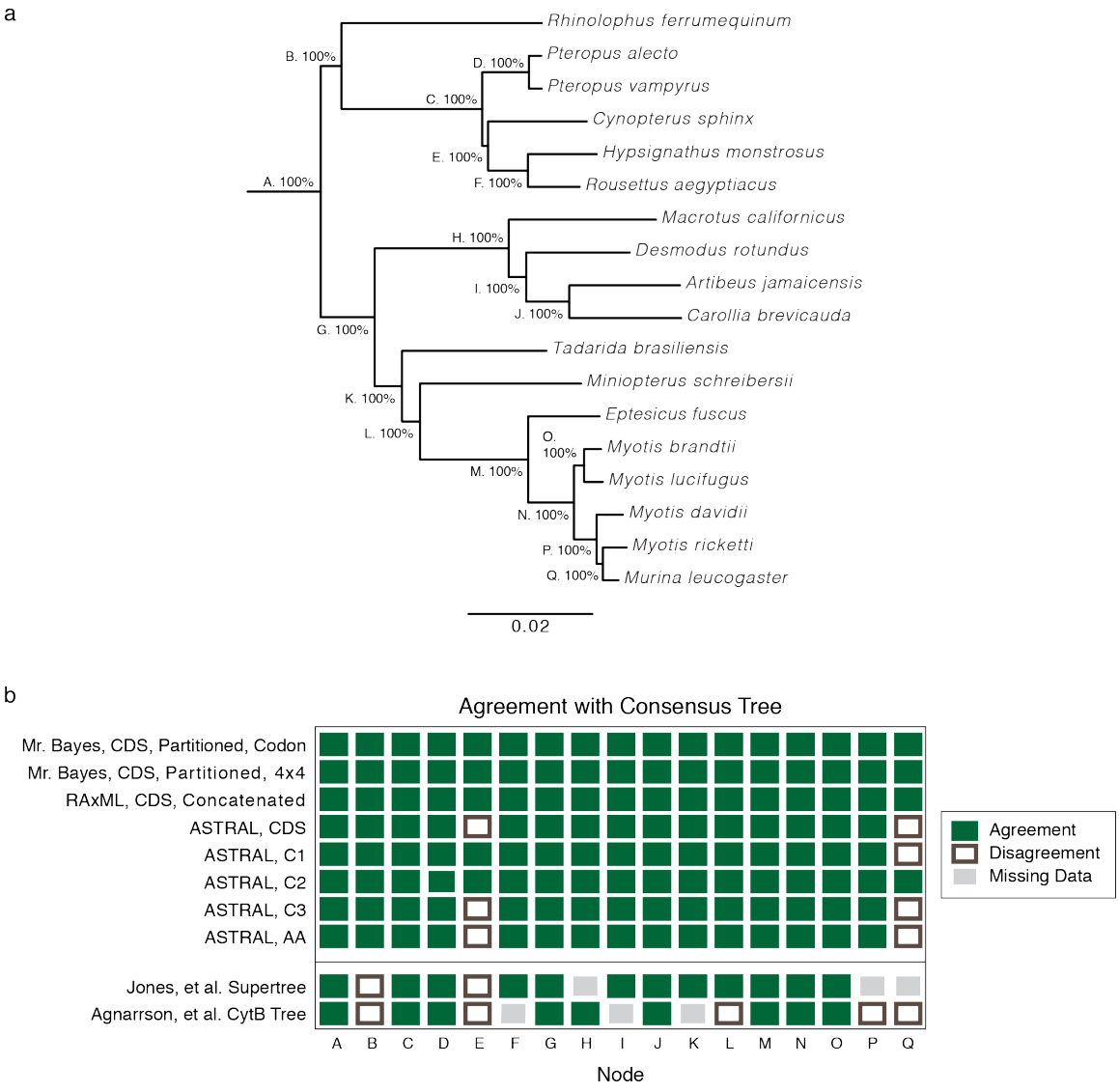


Figure 21. Consensus Chiroptera phylogeny.

Figure 21. Consensus Chiroptera phylogeny.

(a) The consensus phylogeny for the bats considered in this study. Node labels give a name for each node as well as the reported posterior probability for the node under the Mr. Bayes partitioned analysis with full codon model, which agrees at all nodes with the consensus tree, and from which branch lengths are also reported. (b) Comparison of trees constructed by various methods with the consensus tree. For each tree listed on the left, the rectangle at each node indicates agreement or disagreement with the consensus tree on the implied split at the node. Various techniques for building phylogenies from our orthologous gene families are shown in the upper section, comparison with the trees from Jones et al. and Agnarrson et al. in the lower. CDS indicates coding sequence, AA amino acid sequence, C1-3 the coding sequence restricted to those bases in the first, second, or third codon position respectively. All our analyses agree on all species except *C. sphinx* and *M. leucogaster*, the former of which was in three instances placed outside node C, the latter of which was in four instances switched with *M. davidii*. The trees from Jones and Agnarrson both place *R. ferrumequinum* above node G and place *C. sphinx* above node C. Agnarrson also places *M. shreibersii* just above node H and *M. leucogaster* just above node N.

One final species placement of note is *D. rotundus*. The family Phyllostomidae here consists of the bats below node H. Wetterer, et al. proposed that the genus Desmodus be placed sister to the rest of the phyllistomids, which were to have formed the sub-family Phyllostominae[156]. Our placement of *M. californicus* sister to *D. rotundus*, however, disrupts Phyllostominae, in agreement with the tree proposed by Rojas, et al.[157]

Positive Selection Analysis

Finally, we looked for signatures of positive selection, using the PAML software package[158] to estimate dN/dS. To account for effects of gene length on dN/dS calculations, we calculated the maximum value of dN/dS in any 30 amino acid window for each gene (Figure 22). As expected, most genes were under overall purifying selection with maximum dN/dS < 1. However, we found 299 genes with patches of dN/dS > 1, indicating positive selection.

To take a look at what kinds of proteins are under positive selection, we used the GO_MWU package[159] to look at the gene ontology (GO) classifications—molecular functions, cellular components, and biological processes—which are over-represented in the genes with dN/dS > 1 (Figure 23). The molecular function most highly associated with the positively selected genes is receptor binding, primarily of cytokines, and the biological processes associated are thus unsurprisingly dominated by immune responses, response to stimulus, and cell recognition. Such receptors are often coopted for viral invasion into the cell, so identification of these genes will hopefully prove useful to the bat and virology communities.

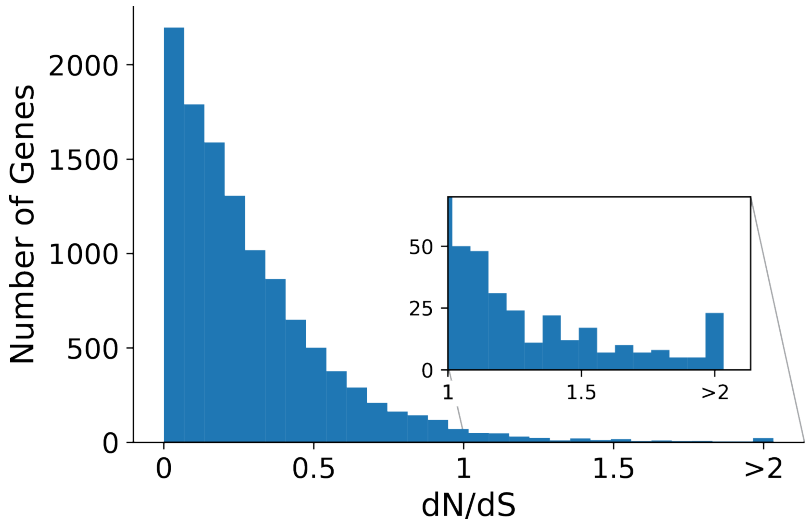


Figure 22. Distribution of dN/dS in all genes.

Maximum dN/dS value for any 30 amino acid patch in each of the 11,572 gene families with ≥ 6 species and ≥ 30 amino acid positions in the MSA.

DISCUSSION

Understanding the molecular history of bats is important not just for the study of Chiropteran zoology, but also for the study of several major zoonotic viruses for which bats provide a viral reservoir. In this study, we set out to look at both the history of speciation and of positive selection in all genes in 18 species of bats. Using both genomic and transcriptomic data, we were able to find 12,611 orthologous gene families. We ourselves provided novel transcriptome data for two of these bats, *Hypsignathus monstrosus* and *Rousettus aegyptiacus*, for which we annotated 7,125 and 8,570 genes respectively. We furthermore developed a novel, general data cleaning method for filtering exons with non-random structural errors, in this case observed to result from genomic vs. transcriptomic data. The MSAs of these gene families, both before and after exon filtering, are available for use by the wider bat and virology communities.

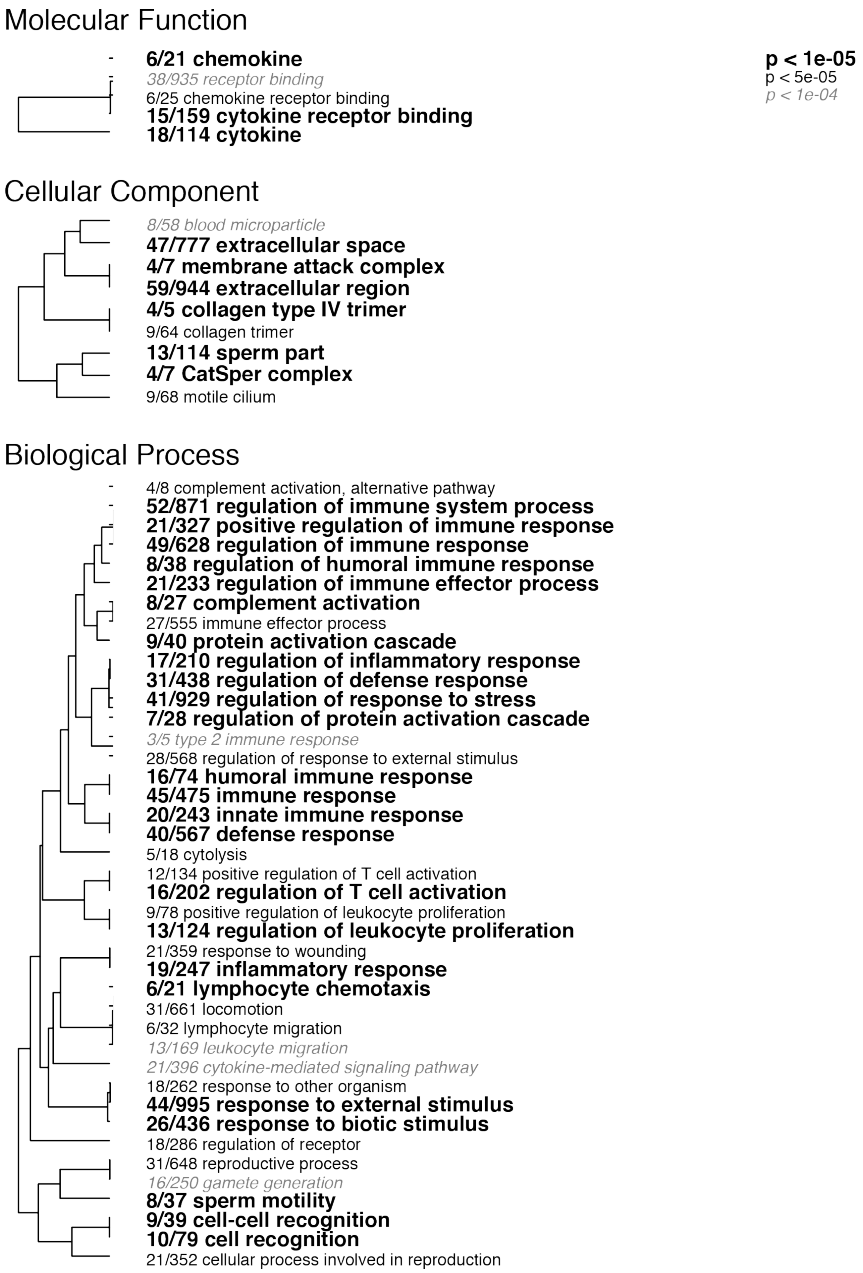


Figure 23. GO categories over-represented in genes under positive selection.

Font size and weight reflects false discovery rate-adjusted p-values. Fractions at the start of each GO category indicate the number of genes under positive selection in that category over the total number of genes in the category. The tree on the left shows the hierarchical relationships between the GO categories within the gene ontology.

Using these orthologous gene families, we were able to reconstruct the phylogeny of the order Chiroptera using multiple methods. Due to the sheer scale of the data, we resolved each node in the tree with 100% reported posterior probability, though the topology differed slightly depending on the analysis method. Our results support the division of Chiroptera into the two suborders Yinpterochiroptera and Yangochiroptera, in disagreement with the traditional division into Mega- and Microchiroptera. We furthermore provide evidence for the placement of *Miniopterus schreibersii,* in which we agree with Hoofer and Bussche, supporting their proposal for the separation of Miniopteridae into its own family[150]. We also provide evidence for the disruption of proposed subfamily Phyllostominae by *Desmodus rotundus*. And most intriguingly, we saw *Murina leucogaster* placed in the Myotis family, which will require further investigation.

We performed positive selection analysis on each orthologous gene family, identifying 299 genes with dN/dS values characteristic of positive selection. Interestingly, these genes were most significantly associated with receptor binding, which could be indicative of a history of viral antagonism. Further study of these genes could shed light on mechanisms for viral entry into and activity inside the cell, including host mechanisms of viral resistance and infection mitigation. We hope identification of these genes will be useful to the virology community for further exploration of bats and their viral antagonists, particularly including those relevant to human health.

METHODS

Sequencing of H. monstrosus and R. aegyptiacus

H. monstrosus RNA was acquired from lung tissue, converted to cDNA, enriched for mRNA via poly-A pull down, and sequenced in two runs on Illumina HiSeqs 2000 and 2500 respectively, both 2x101 bp paired end reads. *R. aegyptiacus* RNA was acquired from

kidney tissue, similarly processed, and sequenced in two runs on Illumina HiSeq 2000 and NextSeq, with 2x101 bp and 2x151 bp reads.

Data Cleaning and Assembly

Sequencing data was first cleaned to remove sequencing adapters and low-quality bases with Trimmomatic, with appropriate settings for each data set. Trinity was run with all reads, all reads with in-silico normalization, and with ~35 million read subsamples for those bats with large data sets as recommended in Francis, et al[160]. Trans-ABySS was run with k-mer lengths of 32, 64, and all multiples of five from 25 to 60.

For genomic data, loci annotated as alternative loci were ignored, as well as readthrough genes. A few instances appeared with isoforms labeled as separate genes; these were manually reduced to one longest isoform.

Ortholog Search

First, we found orthologs between all species using only the genomic datasets, where syntenic evidence helps confirm orthology rather than paralogy (see below). The first step was all-vs-all blasting, filtered for e-values no higher than 10^{-5}. Reciprocal blast hits were considered as tentative ortholog predictions[141]. Tentative predictions were further filtered by including evidence from public ortholog annotation in Biomart, and where available, syntenic evidence. The resulting ortholog prediction graphs for each gene were filtered to remove any connected components with paths connecting two genes from the same species. In the resulting ortholog sets, there were a few cases where genes with the same name in two species were placed in different ortholog prediction groups. In these cases, all incriminated groups were simply removed.

Next, we added orthologs from the transcriptomes. First, genomic transcripts were translated to amino acid sequence and blastx-tblastn reciprocal blast hits were found

between all genomic high-quality orthologous genes and all transcriptome assemblies, again using an e-value cutoff of 10^{-5}. We then created HMMER models of each of the genomic orthologous gene sets and searched within the reciprocal blast hit contigs for the best HMMER hit for each gene, filtering for those hits with e-value below 10^{-10}, extracting only the portion of each contig specified in the HMMER hit. We then filtered all transcripts which differed in length from the median genomic sequence length by more than 25%.

Syntenic Evidence

Whole genome alignments were performed following a procedure described on the UCSC Genome Browser wiki, which for our purposes only required aligning, chaining, and netting. Alignment was performed using Lastz[161], while chaining and netting were performed with kentUtils[143]. Tentative orthologs are considered to have evidence of synteny if they are in syntenic regions, as defined by the top-level nodes of the net.

The algorithm used for determining gene-proximity evidence of synteny is a simple extension of the algorithm in Jarvis, et al[142]. Let species A and species B have genes a_1 and b_1 respectively which are tentatively orthologs. Then let a_2 be the nearest gene to a_1 on the same chromosome which has a tentative ortholog in species B, b_2. If the number of genes between a_1 and a_2 and the number of genes between b_1 and b_2 are both less than 5, then we consider the ortholog pair a_1-b_1 to have syntenic evidence. If there are at least 5 genes in each direction, but the above is not true, then there is evidence against synteny. In the case of not enough genes to either side, it is undetermined.

Multiple Sequence Alignments and Best Genomic Isoforms

Multiple sequence alignments were generated on amino acid sequences with MAFFT L-INS-i, the slower but more accurate version of the popular MAFFT alignment software[162]. Manual inspection revealed that while the HMMER models had quite consistently found

the same isoform from all the transcriptomic data sets, the genomic data were slightly less consistent. This is not surprising, as different species can have different annotated longest isoforms. So, for each gene in each species with genomic data, we went back and found the isoform most similar to the consensus isoform.

The algorithm for finding best splice variant for a gene *g* in a given orthologous set *S* is as follows. First, find the consensus sequence of *S* from the MAFFT L-INS-i alignment. The consensus sequence is simply the identity in each column of the amino acid identity which is found in a majority of transcripts, or X if no single value is the majority. Next, align all splice variants of gene *g* against the consensus sequence, again using MAFFT L-INS-i, and score each by the number of non-gap, non-X positions in agreement with the consensus. Select the splice variant with the highest score. This resulted in improved selection of 3,444 splice variants.

Finally, we realigned for final gene alignments. Corresponding cds alignments were created using pal2nal[163].

Phylogenetic Analyses

Partitioned and unpartitioned analyses in Mr. Bayes otherwise used the same parameters: 2 runs with 4 chains run for 1,000,000 steps sampled every 500 steps and a burnin of 400,000 steps. Gamma models were also subsampled, resulting in a >99% reported posterior probability for the gtrsubmodel[123145] model in both cases. RAxML was run with the rapid bootstrapping and ML algorithm, the GTRGAMMA model, and with 100 different starting trees. Gene trees were created using Mr. Bayes using 20,000 steps sampled every 10 steps with 500 step burnin and an inverse gamma model. Gene tree node support statistics were calculated using dendropy.

Positive Selection Analysis

Positive selection was measured for each gene with PAML using the M8 model and F61 codon model. This model finds an expected dN/dS value for each codon position. We averaged these expected dN/dS values over a sliding window 30 amino acids wide, and reported the maximum such value for each gene.

GO Analysis

GO terms were assigned to each gene using the annotations associated with the gene name from Homo sapien downloaded from the Gene Ontology Consortium[164,165]. These GO annotations, as well as binary positive selection classification for each gene (dN/dS > 1) were input to the GO_MWU R package[159], which collapses very similar GO categories, performs Fisher's exact test for each category, reports false discovery rate-adjusted p-values, and plots results.

Software Versions

Software versions used in this project were Trimmomatic 0.32, Trinity 20140717, Trans-ABySS 1.5.1, BLAST 2.2.29+, HMMER 3.1b2, MAFFT 7.221beta, Lastz 1.02.00, kentUtils 302, Mr. Bayes 3.2.6, RAxML 8.2.6, ASTRAL 4.7.8, PAML 4.8, GO_MWU commit 568e4f5.

Appendices

APPENDIX A: SUPPLEMENTAL FIGURES

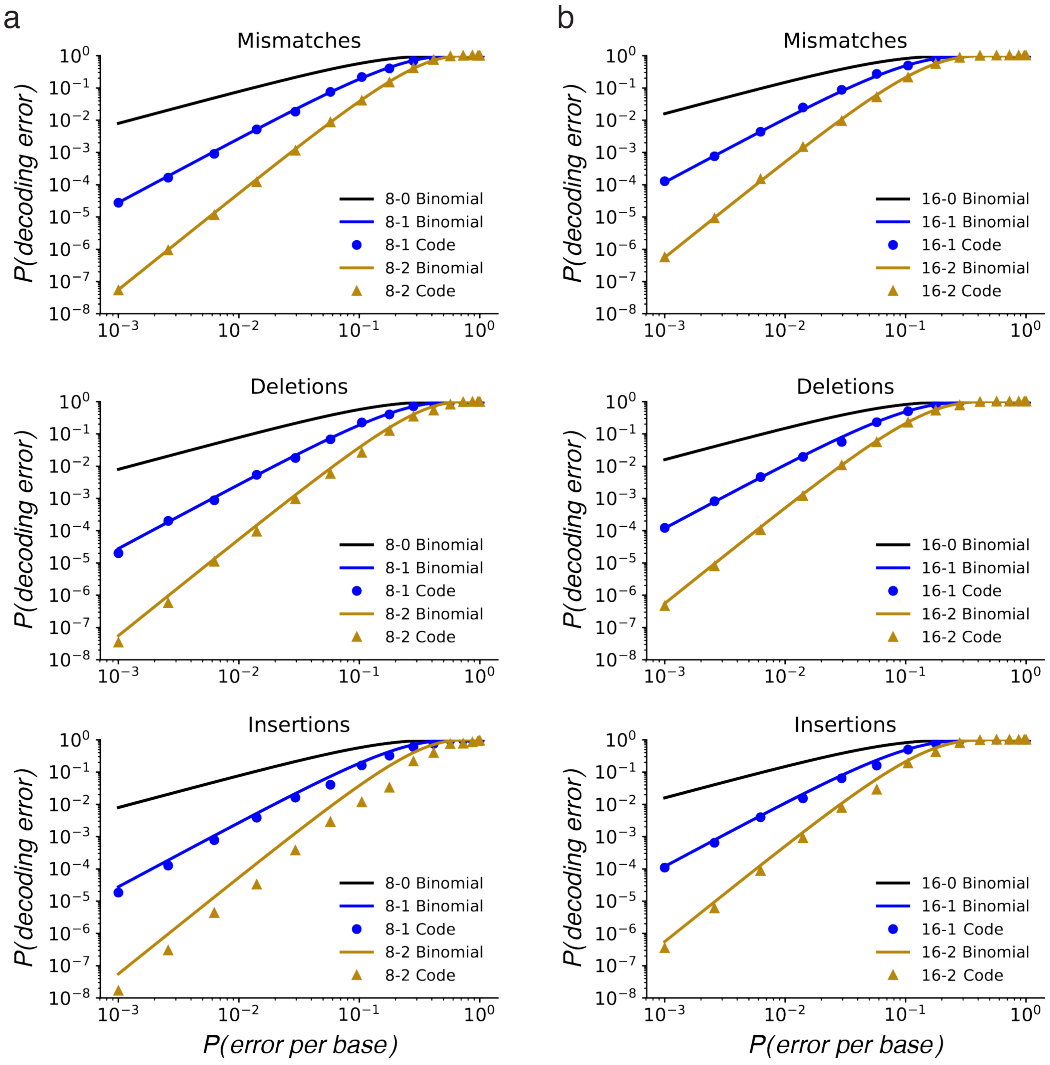


Figure 24. Error rate simulations by error type.

a-b. The simulations performed for Figure 6, repeated for each error type—substitutions, deletions, insertions—individually. Shown for length (a) 8 and (b) 16 barcodes. Barcode sets are labeled according to length and number of errors corrected; for example, the 16-2 code is length 16 and corrects up to 2 errors. Mismatches follow the binomial approximation closely, while deletions and especially insertions perform slightly better than the binomial approximation.

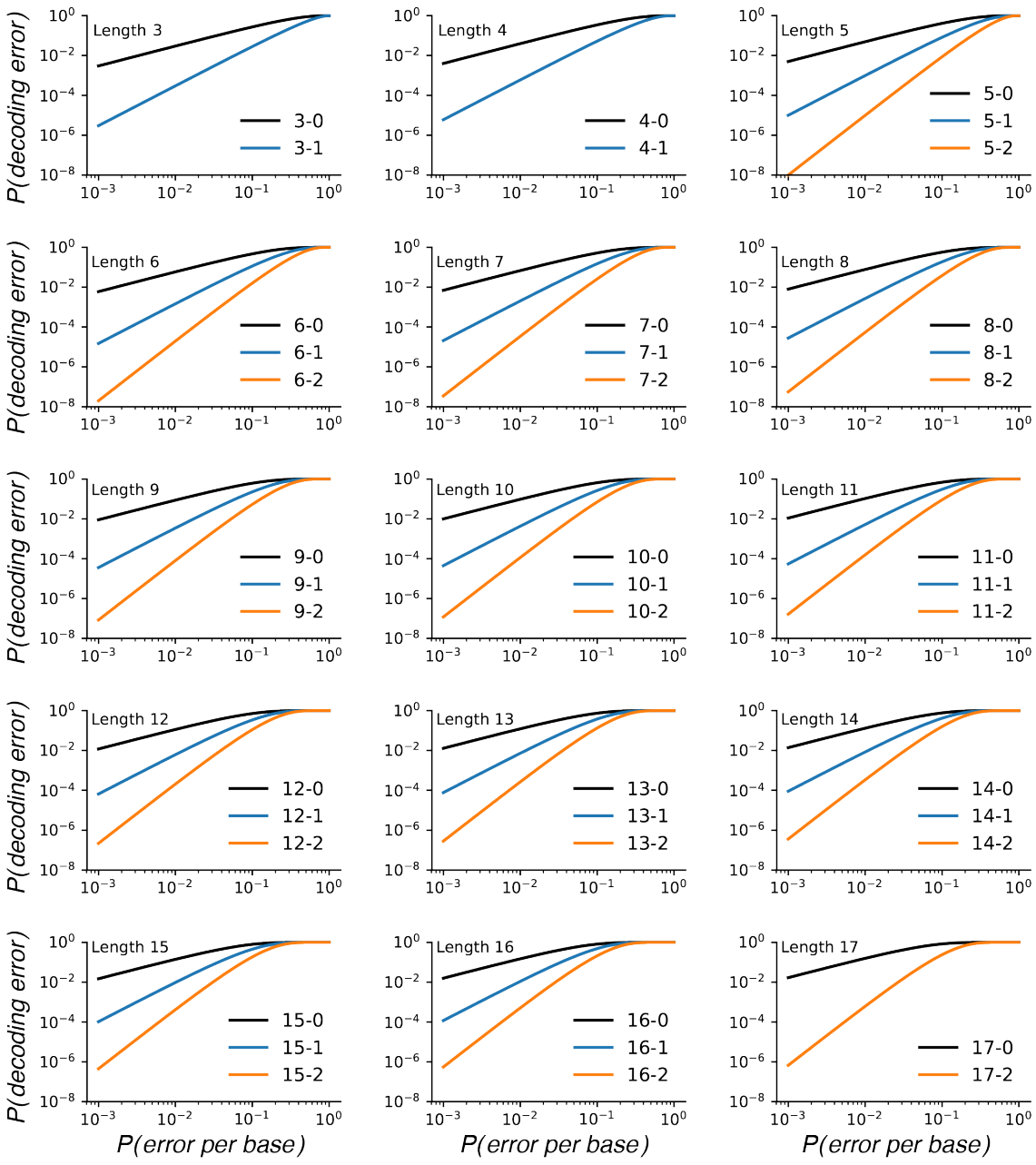


Figure 25. Error rate comparison with constant barcode length.

The binomial approximation of the decode error rate as a function of the error rate per base, grouped by given barcode length.

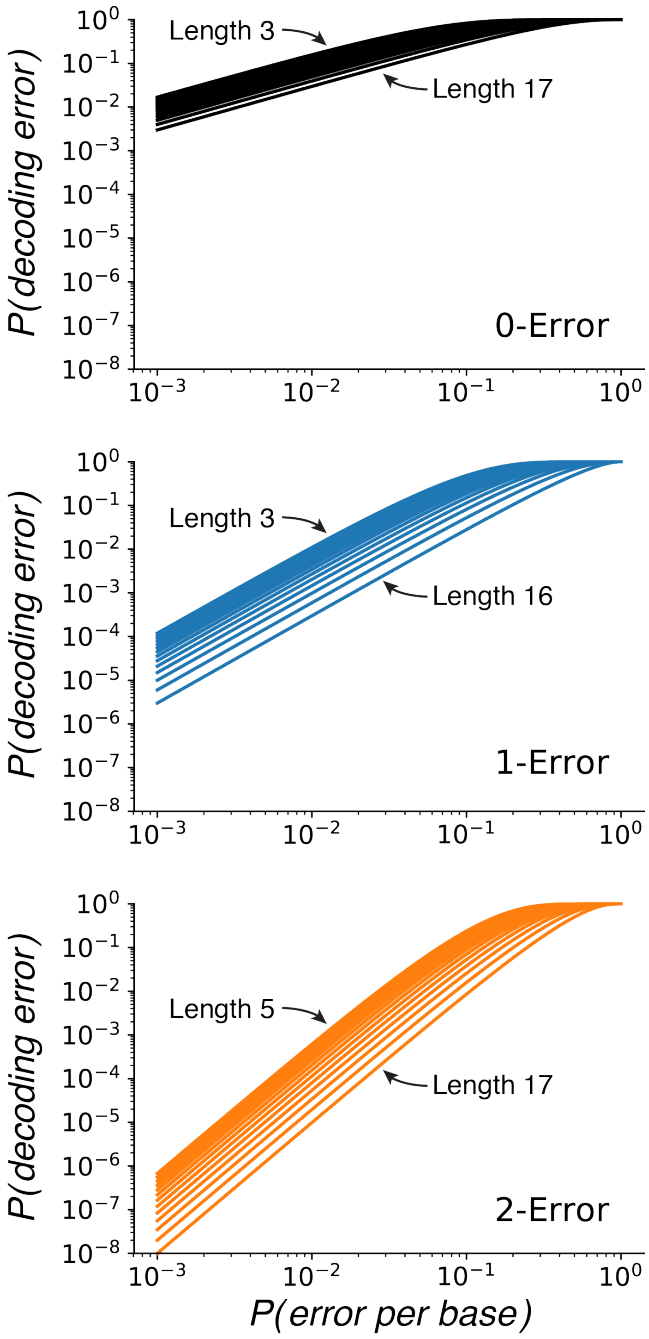


Figure 26. Error rate comparison with constant barcode number of errors corrected.

The binomial approximation of the decode error rate as a function of the error rate per base, grouped by given number of errors corrected.

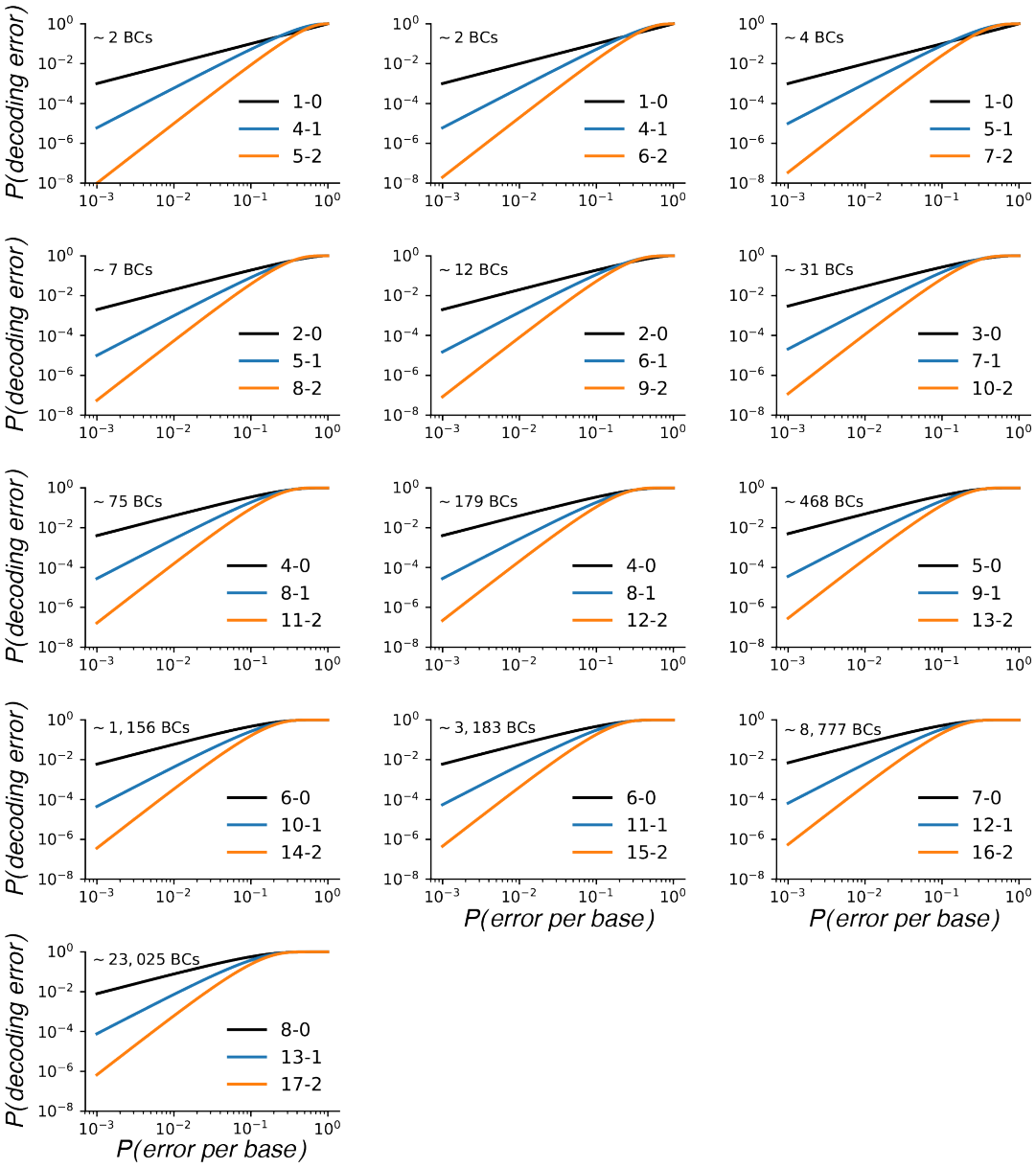


Figure 27. Error rate comparison with constant number of barcodes.

The binomial approximation of the decode error rate as a function of the error rate per base, grouped by number of barcodes. Numbers of barcodes were not precisely equal. Rather, each panel starts with the number of 2-error correcting barcodes and uses the smallest 0- and 1-error correcting barcode sets with at least as many barcodes.

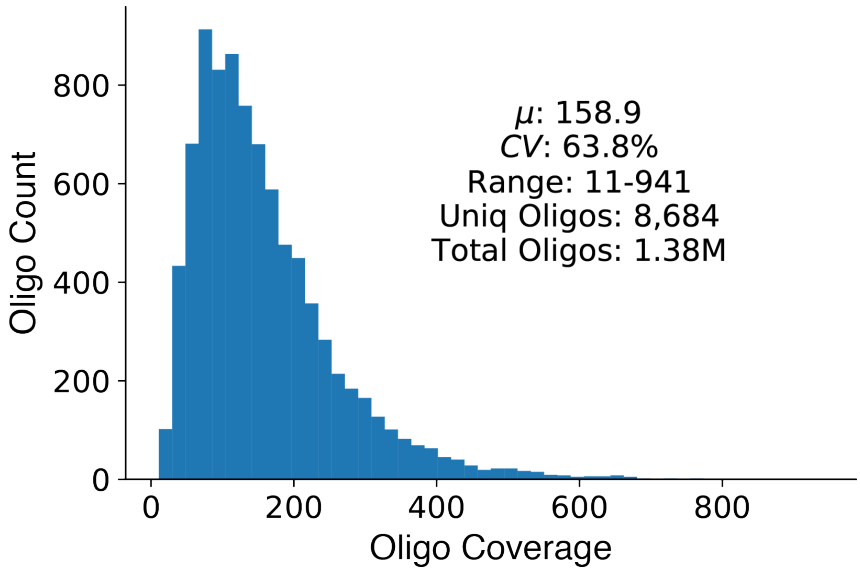


Figure 28. Barcoding experiment sequencing coverage.

Coverage histogram and statistics for the FREE code validation experiment. Each of the 8,684 oligos was observed with average coverage of 159x.

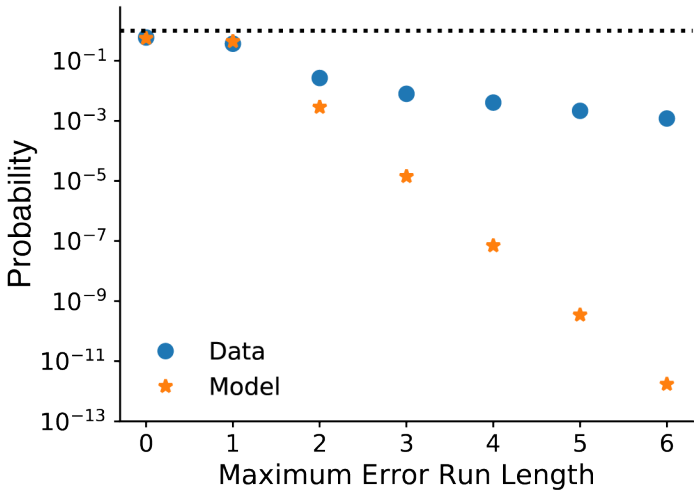


Figure 29. Maximum error run length probabilities.

The probability distribution of maximum consecutive-error run lengths from a model assuming independent errors (Appendix C) and from our data. The two differ significantly because errors in our data are not independent.

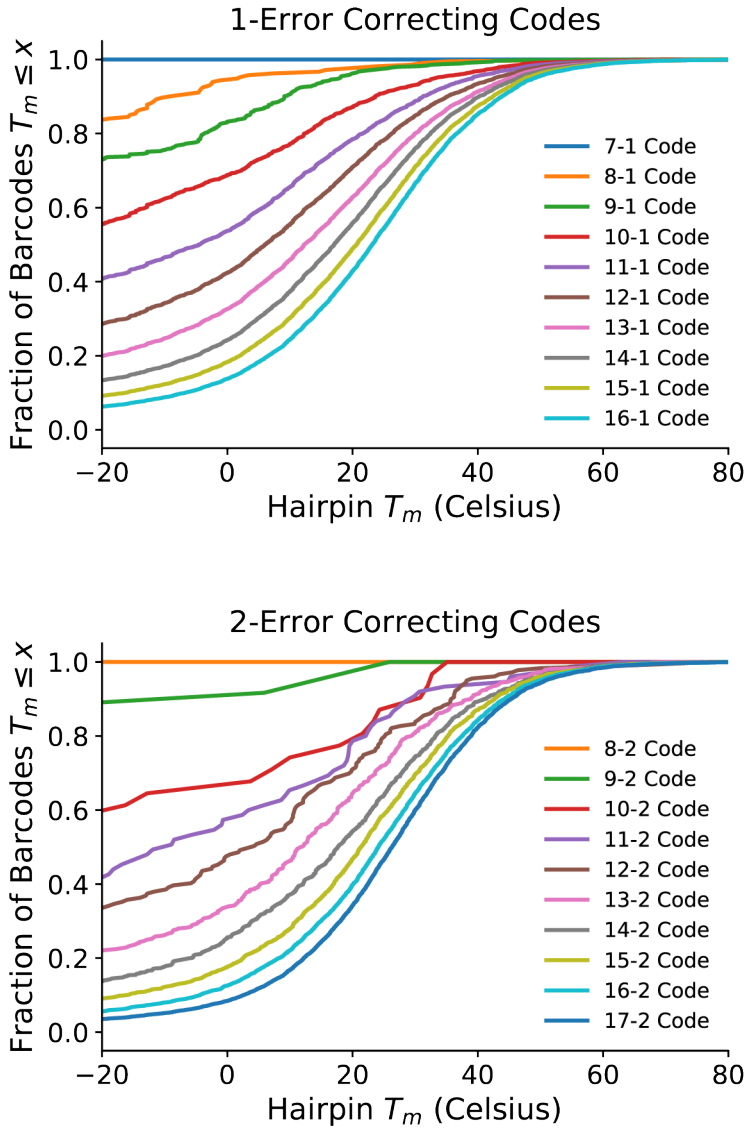


Figure 30. Barcode hairpin melting temperatures.

Hairpin melting temperature CDFs are shown for all barcodes libraries included with this manuscript. The barcodes included here nearly all have Tm < 60◦C, and users can further filter the barcode sets to avoid hairpins in their specific experimental conditions.

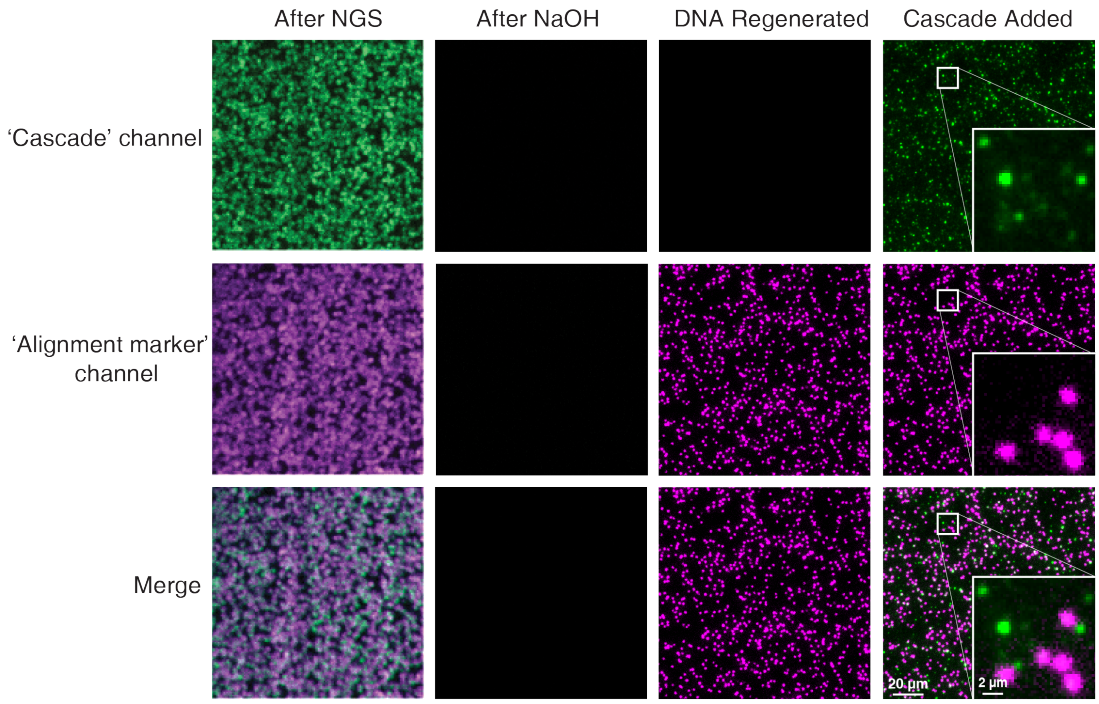


Figure 31. Regenerating DNA clusters on a sequenced MiSeq chip.

After sequencing, the chip contains residual fluorescence in all emission channels (left). The residual fluorescence and sequenced DNA strands are stripped with NaOH and the DNA is regenerated (middle two panels). Finally fluorescent Cascade is incubated in the chip and binds a subset of the DNA clusters (right, green). PhiX clusters are labeled with a fluorescent oligonucleotide (magenta) for downstream image alignment (see Computational Methods).

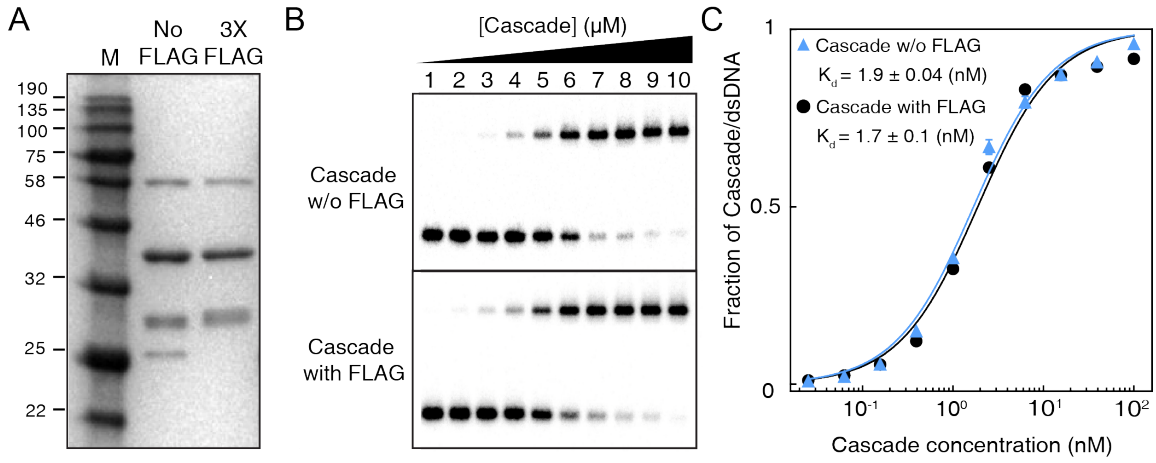


Figure 32. Comparison of Kd values between Cascade with FLAG and without FLAG.

(A) Both Cascade proteins were purified and verified via SDS PAGE gel electrophoresis. EMSA was performed with both Cascade proteins and dsDNAs for "GAAG" PAM sequence and Kd values were calculated. Radiolabeled-dsDNA (0.1 nM) for "GAAG" PAM sequence was incubated with titrated Cascade (1: 0.025 nM, 2: 0.063 nM, 3: 0.16 nM, 4: 0.39 nM, 5: 1 nM, 6: 2.5 nM, 7: 6.3 nM, 8: 16 nM, 9: 39 nM, 10: 100 nM) and resolved with an 2.5 % agarose gel (B) and Kd was calculated by fitting the fraction of Cascade/dsDNA and Cascade concentration (C). Kd values were calculated with duplicate experiments.

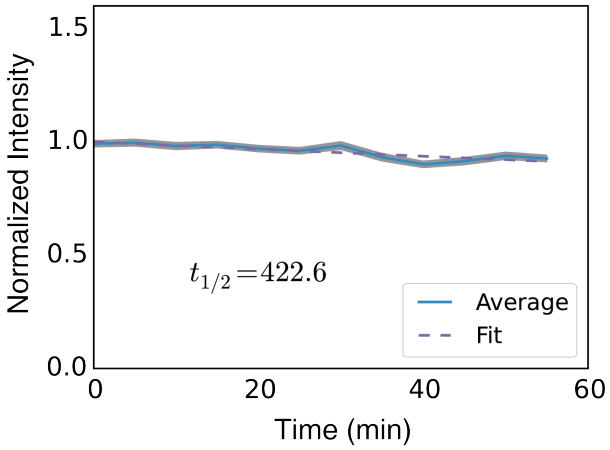


Figure 33. Fluorescent signal loss for Cascade-bound clusters using CHAMP.

10 nM Cascade was incubated on a prepared NGS chip for 10 minutes at 60˚C, then washed and labeled with anti-FLAG Alexa488 antibody. Images were then collected every five minutes for one hour. The intensity of clusters containing ideal target sequence was determined (mean ± SEM), and fit to an exponential decay curve (dashed line).

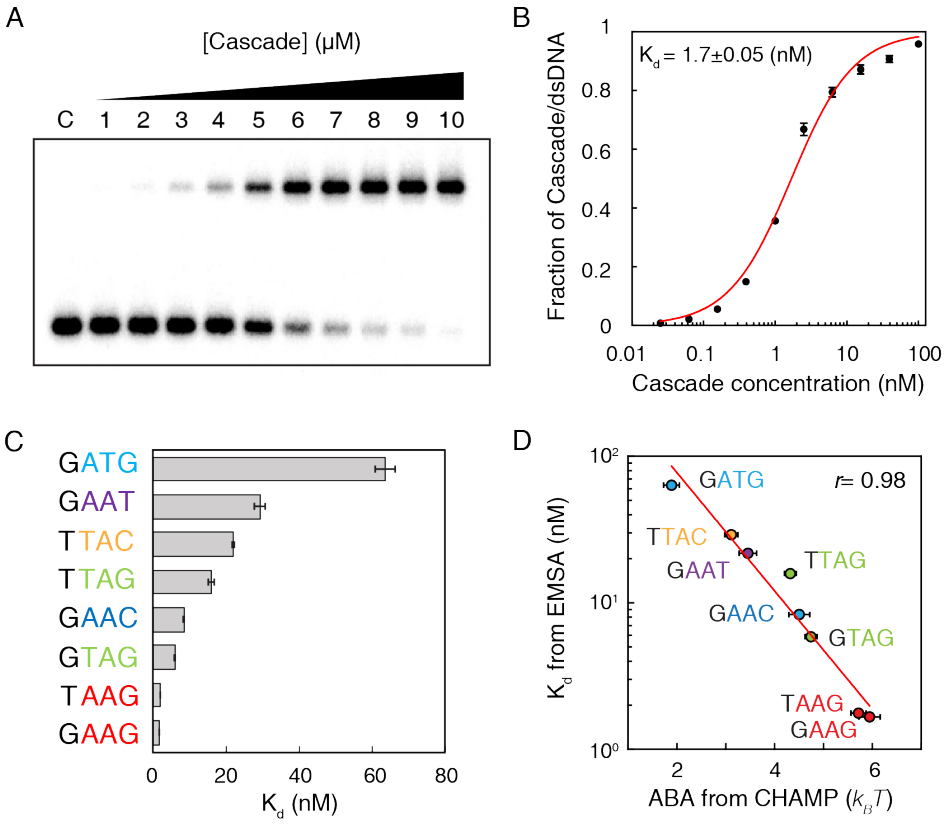


Figure 34. Electrophoretic mobility shift assays (EMSAs) correlate with ABAs.

EMSAs were performed with Cascade and dsDNAs for 8 different PAM sequences and then Kd values were calculated. Radiolabeled-dsDNA (0.1 nM) with a GAAG PAM sequence was incubated with titrated Cascade (C: No Cascade, 1: 0.025 nM, 2: 0.063 nM, 3: 0.16 nM, 4: 0.39 nM, 5: 1 nM, 6: 2.5 nM, 7: 6.3 nM, 8: 16 nM, 9: 39 nM, 10: 100 nM) and (A) resolved with a 2.5 % agarose gel and (B) the Kd was calculated by fitting the titration curve to a Hill equation. (C) Kd was calculated from three replicates as in (B). (D) For 8 PAM sequences, Kd values obtained from EMSA were plotted with ABA values from CHAMP.

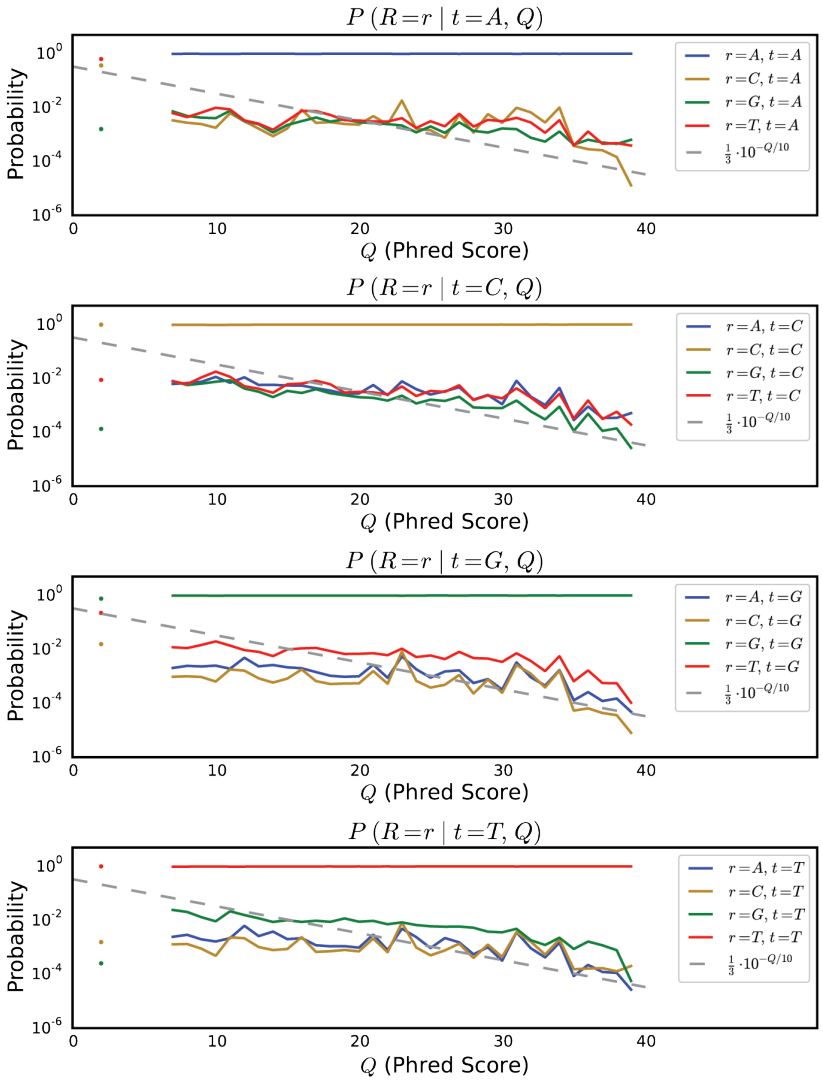


Figure 35. Mapping sequence scores to error probabilities.

Figure 35. Mapping sequence scores to error probabilities.

Information from both reads was used to produce high confidence inferred sequences. I developed a simple Bayesian model for inferring each base, assuming independent errors in each position and a flat prior. For each position, this gives:

$$P(t_i = b \mid R_{1i}, Q_{1i}, R_{2i}, Q_{2i}) \propto P(R_{1i} \mid t_i = b, Q_{1i}) \cdot P(R_{2i} \mid t_i = b, Q_{2i})$$

where i is the position in the aligned sequence, t_i is the true sequence base, b is a base identity (A, C, G, or T), R_{1i} and R_{2i} are the read bases, and Q_{1i} and Q_{2i} are the Phred scores. Maximum *a posteriori* (MAP) values were taken as the inferred sequence. Shown above are all values for $P(R = r \mid t = b, Q)$ observed from 10 billion read bases in PhiX reads mapped without gaps to the Illumina PhiX genome, observed to have the following mutations relative to the NCBI PhiX genome gi|9626372: G587A, G833A, A2731G, C2811T, C3133T. The grey dashed line shows the implied probability for each mismatch given the Phred score, and was used wherever observed values were not available. Base reads other than A, C, G, or T and bases with Phred scores less than or equal to 2, which Illumina reserves for special use, were discarded as missing data.

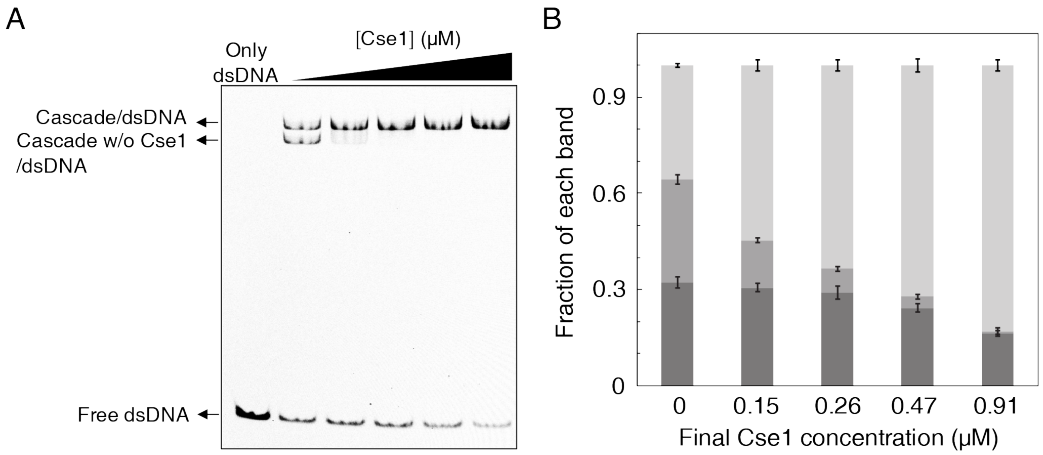


Figure 36. Cse1 dissociates from the Cascade complex.

(A) EMSA of a DNA with a TTAC PAM and perfectly-paired protospacer. Cse1 is dissociated from 50% of the Cascade-DNA complexes (lane 2). Adding excess Cse1 can drive its re-association with the nucleoprotein complex. (B) Quantification of three replicates similar to (A). Light gray: Cascade/dsDNA, gray: Cascade without Cse1, dark gray: free dsDNA. Error bars indicate S.D. [DNA]: 2 nM, [Cascade]: 39 nM, additional [Cse1]: 0, 0.11, 0.22, 0.43 and 0.87 μM. All components were incubated together at 62 °C for 10 min.

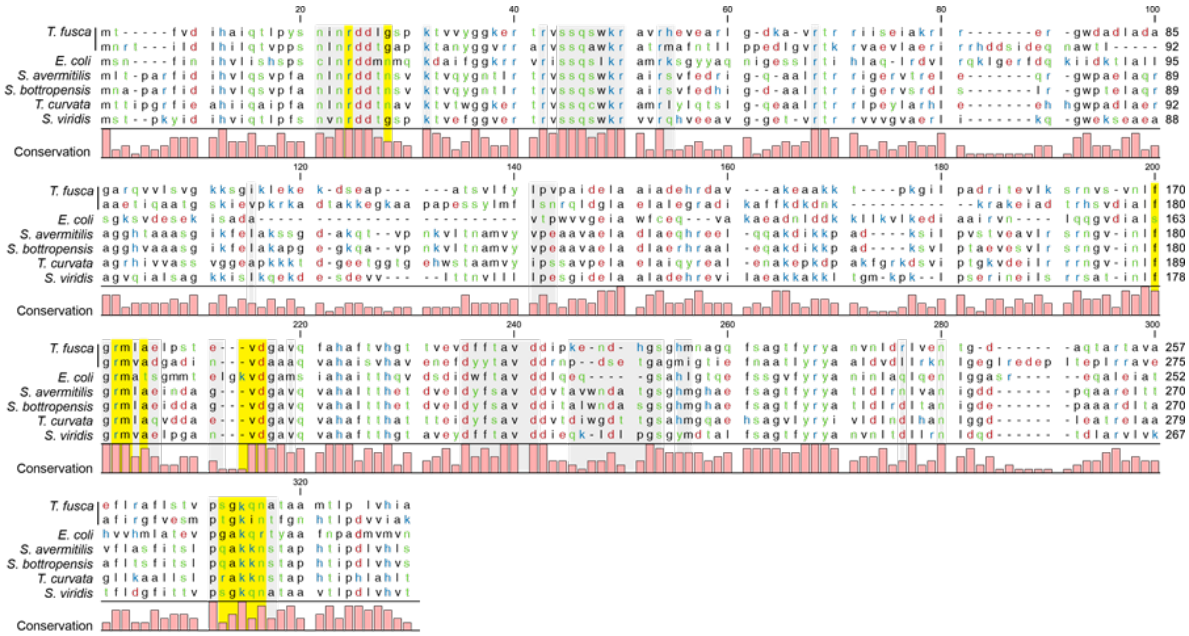


Figure 37. Cas7 sequence alignment.

The crRNA-exposed face of T. fusca Cas7 is highly conserved. Both T. fusca Cas7 sequences were aligned to Cas7s from five distant species including Escherichia coli, Streptomyces avermitilis, Streptomyces bottropensis, Thermonospora curvata, and Seratia viridis. T. fusca encodes two Cas7 variants in two distinct CRISPR operons. Alignment was performed using CLC Sequence Viewer 7.7. Letter coloring indicates polarity (red: acidic, blue: basic, green: polar, black: nonpolar). Pink bars show percent sequence conservation. Grey shading indicates peptides within 6 Å of the crRNA, based on E. coli Cascade in complex with dsDNA (Hayes et al., 2016). Yellow shading identifies peptides within 6 Å of ribonucleotides forming the observed 3-nt periodicity centered between flipped-out bases.

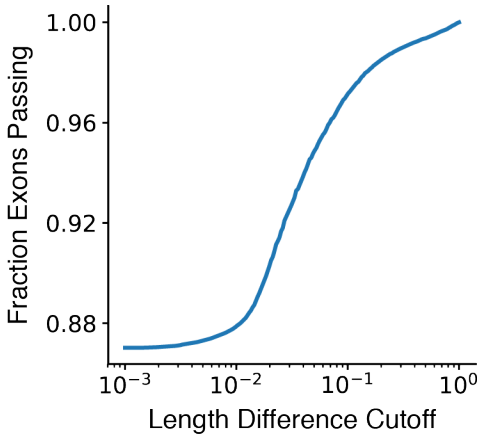


Figure 38. Exons accepted vs. length difference cutoff.

APPENDIX B: SUPPLEMENTARY TABLES

Barcode Length	1-Error Correction	2-Error Correction
3	2	-
4	3	-
5	10	2
6	27	2
7	67	4
8	213	7
9	554	12
10	1,903	31
11	6,161	75
12	17,214	179
13	56,736	468
14	157,197	1,156
15	518,509	3,183
16	1,636,418	8,777
17	-	23,025

Table 2. Numbers of FREE barcodes.

Number of FREE barcodes for each barcode set included with this paper, by barcode length and number of errors corrected.

Species	Assembly Accession Number	Total bp	Scaffold Count	Scaffold N50
Eptesicus fuscus	EptFus1.0	2,026,629,342	6,789	13,454,942
Myotis brandtii	ASM41265v1	2,107,242,811	169,750	3,225,832
Myotis davidii	ASM32734v1	2,059,799,708	101,769	3,454,484
Myotis lucifugus	Myoluc2.0	2,034,575,300	11,654	4,293,315
Pteropus alecto	ASM32557v1	1,985,975,446	65,598	15,954,802
Pteropus vampyrus	Pvam_2.0	2,198,284,804	36,094	5,954,017

Table 3. Genome assembly accession numbers and statistics.

Basic information and statistics for each bat genome assembly, including the genome assembly accession number, total number of base pairs, number of scaffolds, and scaffold N50. N50 is the length of the contig at which half of assembled bases are in contigs of equal or greater length, a standard assembly statistic.

APPENDIX C: FREE BARCODES SUPPLEMENTAL MATERIALS

Sphere iterator

Central to our generation and decoding algorithms is the ability to deterministically iterate over decode and encode spheres. Recursive iteration is far too slow for practical use due to redundancy. For example, attempting to find $DecodeSphere_2(B)$ by finding $DecodeSphere_1(W)$ of all words W in $DecodeSphere_1(B)$ results in iterating over each 2-error word at least twice, by switching the order of added edits. As the number of edits, m, grows, the redundancy grows as $m!$ due to edit permutations.

So, to iterate over a sphere centered at barcode B, we instead built a method to iterate over all words at a given FREE divergence d from B and then iterate over d as needed. We additionally exploit the following identities regarding substitution (sub), insertion (ins), and deletion (del) edits to optimize iteration: sub-del = del, ins-del = del-ins = sub, ins-sub = sub-ins, and in the last position, ins = del = sub. Note that use of these identities assumes we are only interested in solid spheres, so will for example iterate over a sequence at divergence d with an ins-del sequence during the previous $d - 1$ divergence sphere with a sub.

Use of encode spheres

The algorithm used for efficient code generation relies on the fact that if a word W is in $EncodeSphere(B)$, then $DecodeSphere(B)$ and $DecodeSphere(W)$ overlap. That is, there exists a word U such that $U \in DecodeSphere(B)$ and $U \in DecodeSphere(W)$. Let $W \in EncodeSphere(B)$. If $W \in DecodeSphere(B)$, then $U = W$ and by symmetry we are done. Suppose W $6\in$ $EncodeSphere(B)$. By the definition of encode spheres, there exists a filled/truncated right end edit path (FREE path) with at most $2m$ edits from B to W. Let U be the filled or truncated sequence m edits along this path from B. With this choice, $FreeDiv(B,U) \leq m$, where the less than or equal sign is in case of any fill/truncation effects.

Furthermore, there are at most m more edits along the path to W by choosing the fill or truncation for word U to allow use of the same FREE path to W. Then, by use of symmetry, $FreeDiv(W,U) \leq m$, and we are done.

Primer processing

Primers were used both chemically for library amplification and informatically to distinguish left from right sides. However, the possibility of insertions and/or deletions in these primer sites introduced some uncertainty in the starting position of the DNA barcodes. To address this, we wrote a custom adaptation of the Smith-Waterman algorithm for overhanging sequences. The user specifies an expected primer sequence, a full-length observed read, and a maximum allowable number of errors, which we chose to be 2 for both the left (19 bp) and right (18 bp) primers. Using the modified Smith-Waterman algorithm with unity penalties for all error types, we identified the highest scoring prefix of the observed sequence which matches the expected sequence. If two or more possible lengths had the same score, we chose the one closest to the expected length. If the number of edits is less than or equal to 2, this best inferred length then determines the position to be used as the start of the barcode sequence.

Experimental decode errors

Decode errors are detected by whether or not the left and right barcodes, as shown in Figure 5a, match an intended left/right barcode pair. There are two possible ways to decode incorrectly: either by decoding to a wrong barcode or by decoding to "None" if the observed barcode is not in any decode sphere at all. If a barcode decodes to "None", then that decode is obviously an error. If a barcode decodes to an incorrect barcode, then the observed output is that the left and right barcodes mismatch but it is unclear which is actually the decode error. We determine which barcode is in error by measuring the edit distance of the entire oligo against the two possible intended sequences, accepting the one with lowest edit distance. To measure the 0- and 1-Error correction data in Figure 7, we

then measured the edit distance of each observed barcode to the intended barcode using the primer processing algorithm described above.

This analysis resulted in the detection of chimera oligos, oligos with the left side of one intended oligo and the right side of another. Most of the barcodes which decoded to wrong barcodes matched the wrong barcode with zero errors, which was very unexpected. The decode spheres for 17-mer, 2-error correcting codes contain ~10^4 barcodes, of which ~10^2 are 1-error away and exactly 1 is the 0-error wrong barcode (Figure 4a). Furthermore, the wrong barcode with zero errors is the center word, furthest from the sphere boundary and other barcodes. Thus, seeing a barcode decode to a wrong barcode with zero errors should be vanishingly rare compared with 1 and 2 errors. These together imply that we are not observing random errors. We instead appear to be generating chimera oligos. This is likely explained by degeneracy in the spacer region: the spacers are all different, but have stretches of identical sequences around 20 bp long. Incomplete PCR products could then act as primers for this sequence in later rounds of PCR, creating chimera sequences. To correct for these chimeras, we conservatively assumed the distribution of the number of errors in chimera barcodes is the same as that for correct barcodes, though it is likely higher. The observed number of wrong barcodes with zero errors was < 10^4, the approximate size of a decode sphere, so we accepted that as an approximation for how many chimera oligos had barcodes with zero errors. We then used this number and the distribution of correct barcode errors to approximate how many of the wrong barcodes 1-error and 2-errors away from the wrong barcode were chimera oligos. These were then omitted from analysis.

Maximum error run lengths

The error models used in this paper, both the simpler binomial model and that derived above, assume independent errors at each position, understanding that this is an oversimplification. A quick way to see that the errors in our experimental data are definitely

not independent and to show why this impacts our work directly is to check the distribution of maximum error run lengths, i.e., the maximum number of consecutive errors in a given oligo. We consider the binomial model where each position is either an error with probability p or correct with probability $q = 1 - p$. We wish to know the probability that in a sequence of length n the maximum run of errors will be r bases long. This is a well-studied problem, and we use Simpson's solution as presented by Hald [4]. Briefly, let Z_n be the probability that the maximum run of errors in a sequence of length n is at least r bases long and let z_n be the probability that the first run of r errors ends at the n^{th} position. Then

$$Z_n = z_1 + z_2 + \cdots + z_n.$$

For $n < r$, $Z_n = z_n = 0$ trivially, since there are not enough bases, and for $n = r$, $Z_n = z_n = p^r$ since they must all be errors. For $n > r$,

$$z_n = (1 - Z_{n-r-1})qp^r$$

by definition of z_n, since this is the probability that no run of length $\geq r$ in the first $n - r - 1$ bases, then there is a correct base followed by r errors. One can then recursively find Z_{cr+i} for increasing c, resulting in the general formula

$$Z_n = \sum_{c=1}^{\lfloor n-1/r \rfloor} (-1)^{c+1} \left[p^r \binom{n-cr}{c-1} (qp^r)^{c-1} + \binom{n-cr}{c} (qp^r)^c \right].$$

Finally, for fixed n and p, P(maximum run length = r) = $Z_n(r) - Z_n(r+1)$, shown in Figure 29 using our oligo length, $n = 116$, and measured probability of error, $p = 0.005$ (Figure 5). Our experimental data are similar to this model for maximum runs of zero and one errors, but deviate significantly for maximum runs of more than one error because our errors are not, in fact, independent. This helps explain the deviation of our experimental barcode decoding error rates from the model predictions in Figure 7, since our experimental errors

clump together, increasing the probability of having more than two, say, in a single barcode.

APPENDIX D: CHAMP EXPERIMENTAL PROCEDURES

Protein Cloning and Purification

T. fusca Cascade and Cas3 were over-expressed and purified as described previously[87]. Briefly, the Cascade complex and crRNA were expressed from pET-based plasmids that were co-transformed into BL21 star (DE3) cells (Thermo-Fisher). Cse1 contained a His_6/Twin-Strep/SUMO N-terminal fusion, while Cas6 contained an N-terminal triple FLAG epitope for fluorescent labeling. Single colonies were used to inoculate LB + Kanamycin/Carbenicillin/Streptomycin media. At OD_{600} 0.8, cells were induced with 1 mM IPTG overnight at 25°. Cells were then lysed in 20 mM HEPES [pH 7.5], 500 mM NaCl, 2 μg mL^{-1} DNase (GoldBio) and 1x HALT protease inhibitor (Thermo-Fisher), and the clarified lysate was applied to a hand-packed Strep-Tactin Superflow gravity column (IBA Life Sciences) for purification via the Twin-Strep tagged Cse1. The Cascade complex was eluted with 20 mM HEPES [pH 7.5], 500 mM NaCl, 5 mM desthiobiotin, and then concentrated by centrifugal filtration (30 kDa Amicon, Millipore). The concentrate was then incubated overnight at 4°C with 3.3μM SUMO protease to remove tags from Cse1. The complex was further fractionated over a HiLoad 16/600 Superdex 200 column (GE Healthcare) equilibrated in storage buffer (10 mM Tris-HCl pH 7.5, 150 mM NaCl, 5 mM DTT). Fractions containing the full Cascade complex were determined by SDS-PAGE, pooled, and concentrated to ~5-10 μM (30 kDa centrifuge concentrators, Millipore). Small aliquots were flash frozen in liquid nitrogen and stored at -80˚C.

Antibodies

Cascade and Cas3 were fluorescently labeled with mouse anti-FLAG M2 (F3165, Sigma) and Rabbit anti-HA (RHGT-45A-Z, ICL labs), respectively. Antibodies were conjugated to Alexa488 or Alexa647 at a ratio of ~1:3 antibody:dye according to the manufacturer's instructions (Alexa Fluor antibody labeling kits, Thermo Fisher Scientific). The antibody to dye conjugation ratio was measured using a NanoDrop (Thermo Fisher Scientific)

according to the manufacturer-provided protocol. Fluorescent antibodies were stored in PBS buffer (pH 7.2, with 2 mM sodium azide) at -20 °C.

DNA libraries

All oligonucleotides were purchased from IDT or IBA. Pooled custom DNA libraries were purchased from CustomArray. To profile off-target Cascade recruitment, we designed custom DNA libraries containing a randomized target DNA sequence (e.g., protospacer adjacent motif (PAM) and/or protospacer). A synthetic oligonucleotide with six randomized bases was purchased from IDT and used to profile the extended six nucleotide PAM. To measure the effects of mismatches along the entire sequence, we used "doped" libraries where a given position in the library contained the starting sequence at 91% frequency, and each of the other three nucleotides at 3% frequency each (3% doping). This doping frequency was chosen to provide comprehensive coverage for sequence variants with a Hamming distance less than three on a typical MiSeq chip (representing ~20-25 million unique reads). Insertions and deletions were measured on libraries designed by pooled oligonucleotide synthesis (CustomArray).

Chip regeneration and addition of alignment markers

Custom DNA libraries were sequenced on a MiSeq (Illumina) using a 2x75 or a 2x300 paired end reagent kit (v3). After sequencing, MiSeq chips were stored at 4˚C in TE buffer (10 mM Tris-Cl pH 8.0, 1 mM EDTA). All imaging and chip regeneration steps were carried out in a custom-built microscope stage adapter with integrated microfluidic interconnects. Detailed blueprints of all components are also available via GitHub (https://github.com/finkelsteinlab/champ). Temperature was controlled by PiWarmer, a home-built Raspberry Pi-controlled heating element. PiWarmer was also used to run the heating and cooling cycles required for cluster regeneration. Schematics and code for assembling the temperature controller, as well as protocols for chip regeneration are available via GitHub (https://github.com/finkelsteinlab/piwarmer).

To regenerate the DNA clusters, the MiSeq chip was first washed with 0.1 N NaOH for 5 minutes, followed by flowing TE buffer to neutralize the chip surface for 5 minutes. A flow rate during a whole experiment is fixed at a 100 μl/min. Washing with NaOH also removes residual fluorescent dyes that remain adsorbed to the chip surface after NGS (see Figure 31). After denaturation, the chip was heated to 85˚C and incubated with 500 nM of a user primer (UP) in hybridization buffer (75 mM Trisodium Citrate, pH 7.0, 750 mM NaCl, 0.1% Tween-20). The UP primer was annealed at 85°C for 5 min, followed by ramping down to 40°C for 40 min and then washed with a washing buffer (4.5 mM Trisodium Citrate, pH 7.0, 45 mM NaCl, 0.1% Tween-20) at 40°C for 10 min. The UP primer binds to all user clusters but does not target phiX clusters. The UP primer was extended in a 1X isothermal amplification buffer (20 mM Tris-HCl, pH 8.8, 10 mM $(NH_4)_2SO_4$, 50 mM KCl, 2 mM $MgSO_4$, 0.1% Tween-20) containing a 0.08 U/ul of Bst 2.0 WarmStart DNA polymerase (New England Biolabs) and 0.8 mM of dNTPs by incubating at 60 °C for 10 min. Bst 2.0 WarmStart DNA polymerase was flushed out by washing the chip with a hybridization buffer for 5 minutes at 60 ˚C. Finally, we annealed a phiX primer labeled with Atto647 or Cy3 (atto647-PCP / Cy3-PCP) using a hybridization buffer with the same reaction condition as described above. The resultant fluorescent phiX clusters were used for aligning the FASTQ points to imaged clusters (see Figure 31).

Fluorescence microscopy

All fluorescence images were collected using a Nikon Ti-E microscope in a prism-TIRF configuration equipped with a Prior H117 motorized stage. Each sequenced MiSeq chip (Illumina) was loaded into the microscope stage adapter described above prior to imaging. The chip was illuminated with 488 nm (Coherent), 532 nm (Ultralasers), or 633 nm (Ultralasers) lasers through a quartz prism (Tower Optical Co.). To minimize spatial drift, the microscope was assembled on a floating optical table (TMC). Data were collected with

a 100 ms exposure through a 60X water-immersion objective (1.2NA, Nikon) paired with (i) a quad-band filter (89401 Chroma), a 638 nm dichroic beam splitter, and either a 600 nm long-pass filter or 500 nm long pass / 600 nm short pass filters (Chroma), or (ii) a dual-band filter (ZET532/660m Chroma), a 640 nm dichroic beam splitter, and either a 655 nm long-pass filter or ET4585/65m band pass filter (Chroma), which allowed multi-channel detection through two EMCCD cameras (Andor iXon DU897, cooled to -80°C). Images were collected using Micro-Manager Open Source Microscopy software[166] and saved in an uncompressed TIFF file format for later analysis via a custom written image-processing pipeline (see below).

CHAMP assays

Increasing concentrations of the Cascade complex (0.063, 0.16, 0.39, 1, 2.5, 6.3, 16, 39, 100, 250, and 630 nM) were injected into a regenerated MiSeq chip and incubated at 60 °C for 10 min in imaging buffer (40 mM Tris-HCl, pH 8.0, 150 mM NaCl, 2 mM $MgCl_2$, 1 mM DTT, 0.2 mg ml^{-1} BSA, 0.1% Tween-20). After the incubation, excess Cascade was rapidly flushed out while the DNA-bound proteins were fluorescently labeled by injecting a 100 μl of a 20 nM solution of fluorescently-conjugated anti-FLAG antibodies at room temperature. Control experiments that omitted Cascade indicated that the fluorescent antibody did not bind to the chip surface.

For each Cascade concentration, we imaged up to 812 fields of view spanning nearly 50% of the total sequenced MiSeq chip surface area (Prior ProScan II). The chip was illuminated with 20, 40 or 30 mW of laser power at 488, 532, or 633 nm, respectively (measured at the front face of the TIRF prism). To prevent photobleaching, the lasers were shuttered between subsequent fields of view (Vincent Associates) during the ~15 minutes of image acquisition. No appreciable Cascade dissociation or cluster photobleaching occurred during this time (see Figure 33). In order to avoid pixel saturation at high protein concentrations, ten 100 ms images were captured at each field of view. These images were

summed into a final image and stored in hdf5 files by channel and position. Care was taken to minimize experiment-to-experiment variation by acquiring all concentrations of a titration series in a single day. Following each experiment, the MiSeq chips were de-proteinized with 32 units of Proteinase K (New England Biolabs) in washing buffer overnight at 25˚C and the chip showed no sign of degradation even after twelve Proteinase K treatments. The DNA in a chip could be denatured and re-synthesized up to five times using the regeneration protocol described above.

Electrophoretic mobility shift assay (EMSA)

All EMSAs were performed with radioactively or fluorescently labeled PCR products containing the indicated PAM and protospacer, as well as flanking sequences used in the CHAMP experiments (i.e., Illumina adapters). PCR was performed using 1 ng of template plasmid containing the desired PAM/protospacer, 500 nM of P5 primer for radioactive-labeling or Cy5-P5 primer for fluorescent-labeling, 500 nM of UP primer, 200 μM of dNTPs and 0.5 unit of Q5 high-fidelity DNA polymerase (New England Biolabs) in a 25 μl reaction on an MJ Research PTC-200 Thermal Cycler. The PCR product was purified using a PCR purification kit (Qiagen) and quantified on a Nanodrop spectrophotometer (Thermo Fisher Scientific). For radioactive assays, PCR products were labeled with γ32P-ATP (PerkinElmer) using T4 polynucleotide kinase (New England Biolabs). The labeled PCR products were purified with MicroSpin G-25 columns (GE Healthcare).

Cascade binding assays were performed by incubating 0.1 nM of 32P-labeled dsDNA with increasing Cascade concentrations (0.025, 0.063, 0.16, 0.39, 1, 2.5, 6.3, 16, 39, 100, 250, 630 nM) for 30 min at 62°C in binding buffer. The reactions were resolved on a 2.5% agarose gel run with 0.5X TBE buffer. Gels were dried and DNA was visualized using a Typhoon scanner (GE Healthcare). ImageQuant software (GE Healthcare) was used to quantify the bound and unbound DNA amounts. The fraction of bound DNA was fit to the Hill equation to obtain K_d values. All experiments were repeated in triplicate.

To observe Cas3 binding, Cascade (39 nM) and target dsDNA were pre-bound for 30 min at 62°C in a binding buffer. Then, Cas3 and AMP-PNP (Sigma) were added into the EMSA reaction for final concentrations of 1.1 uM and 2 mM, respectively and incubated for 10 min at 62°C. The reactions were resolved on a 5% native PAGE gel containing 0.5X TBE buffer and visualized using a Typhoon scanner (GE Healthcare).

Cas3 nuclease assays

Cascade (39 nM) was first incubated with Cy5-labeled target dsDNA (2 nM) for 30 min at 62°C in a binding buffer. Then, Cas3, $CoCl_2$ (Sigma) and ATP (Sigma) were added into the EMSA reaction at final concentrations of 650 nM, 111 μM and 1.9 mM, respectively and incubated for 30 min at 62°C. The reaction was quenched with 50 mM EDTA and deproteinized with proteinase K. The reactions were resolved on a 10% denaturing PAGE gel containing 0.5X TBE buffer and visualized using a Typhoon scanner (GE Healthcare).

Plasmid loss assays

The Cascade expression construct was generated by insertion of the Cascade gene cassette (encoding all protein subunits) into a pBAD (ApR) vector. The pre-crRNA expression cassette containing five identical CRISPR units for target A was cloned into the pACYC-Duet-1 (CmR) vector. A 127-bp fragment containing a protospacer and a PAM for target A was cloned into the pCDF-Duet-1 (SmR) vector to serve as the target DNA. All plasmids were sequence verified. In *vivo* assays were performed with *T. fusca* Cascade and Cas3 plasmids as described previously[87].

References

1. Human Genome Project Completion: Frequently Asked Questions. *National Human Genome Research Institute (NHGRI)* Available at: https://www.genome.gov/11006943/Human-Genome-Project-Completion-Frequently-Asked-Questions. (Accessed: 18th March 2018)
2. Schaller, R. R. Moore's law: past, present and future. *IEEE Spectr.* **34,** 52–59 (1997).
3. Johnson, D. S., Mortazavi, A., Myers, R. M. & Wold, B. Genome-Wide Mapping of in Vivo Protein-DNA Interactions. *Science* **316,** 1497–1502 (2007).
4. The ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature* **489,** 57–74 (2012).
5. Miller, M. R., Dunham, J. P., Amores, A., Cresko, W. A. & Johnson, E. A. Rapid and cost-effective polymorphism identification and genotyping using restriction site associated DNA (RAD) markers. *Genome Res.* **17,** 240–248 (2007).
6. Hussmann, J. A., Patchett, S., Johnson, A., Sawyer, S. & Press, W. H. Understanding Biases in Ribosome Profiling Experiments Reveals Signatures of Translation Dynamics in Yeast. *PLOS Genet.* **11,** e1005732 (2015).
7. Langmead, B., Trapnell, C., Pop, M. & Salzberg, S. L. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.* **10,** R25 (2009).
8. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics* **25,** 1754–1760 (2009).
9. Luo, R. *et al.* SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. *GigaScience* **1,** 18 (2012).
10. Zerbino, D. R. & Birney, E. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res.* **18,** 821–829 (2008).

11. Simpson, J. T. *et al.* ABySS: A parallel assembler for short read sequence data. *Genome Res.* **19,** 1117–1123 (2009).

12. Gnerre, S. *et al.* High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proc. Natl. Acad. Sci. U. S. A.* **108,** 1513–1518 (2011).

13. Ronquist, F. & Huelsenbeck, J. P. MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics* **19,** 1572–1574 (2003).

14. Stamatakis, A. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics* btu033 (2014). doi:10.1093/bioinformatics/btu033

15. Mirarab, S. *et al.* ASTRAL: genome-scale coalescent-based species tree estimation. *Bioinformatics* **30,** i541–i548 (2014).

16. Klein, A. M. *et al.* Droplet Barcoding for Single-Cell Transcriptomics Applied to Embryonic Stem Cells. *Cell* **161,** 1187–1201 (2015).

17. Macosko, E. Z. *et al.* Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell* **161,** 1202–1214 (2015).

18. Zheng, G. X. Y. *et al.* Haplotyping germline and cancer genomes with high-throughput linked-read sequencing. *Nat. Biotechnol.* **34,** 303–311 (2016).

19. Kitzman, J. O. Haplotypes drop by drop. *Nat. Biotechnol.* **34,** 296–298 (2016).

20. Haque, A., Engel, J., Teichmann, S. A. & Lönnberg, T. A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications. *Genome Med.* **9,** 75 (2017).

21. Zilionis, R. *et al.* Single-cell barcoding and sequencing using droplet microfluidics. *Nat. Protoc.* **12,** 44–73 (2017).

22. Spies, N. *et al.* Genome-wide reconstruction of complex structural variants using read clouds. *Nat. Methods* **14,** 915–920 (2017).

23. Eroshenko, N., Kosuri, S., Marblestone, A. H., Conway, N. & Church, G. M. Gene Assembly from Chip-Synthesized Oligonucleotides. *Curr. Protoc. Chem. Biol.* **2012,** (2012).

24. Plesa, C., Sidore, A. M., Lubock, N. B., Zhang, D. & Kosuri, S. Multiplexed gene synthesis in emulsions for exploring protein functional landscapes. *Science* eaao5167 (2018). doi:10.1126/science.aao5167

25. Fan, R. *et al.* Integrated barcode chips for rapid, multiplexed analysis of proteins in microliter quantities of blood. *Nat. Biotechnol.* **26,** 1373–1378 (2008).

26. Ma, C. *et al.* A clinical microchip for evaluation of single immune cells reveals high functional heterogeneity in phenotypically similar T cells. *Nat. Med.* **17,** 738–743 (2011).

27. Zimmermann, G. & Neri, D. DNA-encoded chemical libraries: foundations and applications in lead discovery. *Drug Discov. Today* **21,** 1828–1834 (2016).

28. Melkko, S., Scheuermann, J., Dumelin, C. E. & Neri, D. Encoded self-assembling chemical libraries. *Nat. Biotechnol.* **22,** 568 (2004).

29. Kosuri, S. & Church, G. M. Large-scale de novo DNA synthesis: technologies and applications. *Nat. Methods* **11,** 499–507 (2014).

30. Petrone, J. DNA writers attract investors. *Nat. Biotechnol.* **34,** 363–364 (2016).

31. Litovchick, A. *et al.* Encoded Library Synthesis Using Chemical Ligation and the Discovery of sEH Inhibitors from a 334-Million Member Library. *Sci. Rep.* **5,** (2015).

32. CustomArray, Inc. - maker of custom microarrays, oligo pools and instrumentation. Available at: http://www.customarrayinc.com/aboutus_main.htm. (Accessed: 8th January 2018)

33. Peterson, W. W. & Weldon, E. J. *Error-correcting Codes*. (MIT Press, 1972).

34. MacWilliams, F. J. & Sloane, N. J. A. *The Theory of Error-correcting Codes*. (Elsevier, 1977).

35. Lyons, E., Tremmel, G., Sheridan, P., Miyano, S. & Sugano, S. Large-scale DNA Barcode Library Generation for Biomolecule Identification in High-throughput Screens. *Sci. Rep.* **7,** 13899 (2017).

36. Erlich, Y. & Zielinski, D. DNA Fountain enables a robust and efficient storage architecture. *Science* **355,** 950–954 (2017).

37. Levenshtein, V. I. Binary codes capable of correcting deletions, insertions, and reversals. in *Soviet physics doklady* **10,** 707–710 (1966).

38. Costea, P. I., Lundeberg, J. & Akan, P. TagGD: Fast and Accurate Software for DNA Tag Generation and Demultiplexing. *PLOS ONE* **8,** e57521 (2013).

39. Needleman, S. B. & Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **48,** 443–453 (1970).

40. Buschmann, T. & Bystrykh, L. V. Levenshtein error-correcting barcodes for multiplexed DNA sequencing. *BMC Bioinformatics* **14,** 272 (2013).

41. Houghten, S. K., Ashlock, D. & Lenarz, J. Construction of Optimal Edit Metric Codes. in *2006 IEEE Information Theory Workshop - ITW '06 Chengdu* 259–263 (2006). doi:10.1109/ITW2.2006.323799

42. Quail, M. A. *et al.* A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. *BMC Genomics* **13,** 341 (2012).

43. Hamming, R. W. Error detecting and error correcting codes. *Bell Labs Tech. J.* **29,** 147–160 (1950).

44. Huffman, W. C. & Pless, V. *Fundamentals of Error-Correcting Codes*. (Cambridge University Press, 2010).

45. Lee, D. F., Lu, J., Chang, S., Loparo, J. J. & Xie, X. S. Mapping DNA polymerase errors by single-molecule sequencing. *Nucleic Acids Res.* **44,** e118–e118 (2016).

46. Markham, N. R. & Zuker, M. UNAFold: software for nucleic acid folding and hybridization. *Bioinforma. Struct. Funct. Appl.* 3–31 (2008).

47. Zanten, A. J. van. Lexicographic Order and Linearity. *Des. Codes Cryptogr.* **10,** 85–97 (1997).

48. Jung, C. *et al.* Massively Parallel Biophysical Analysis of CRISPR-Cas Complexes on Next Generation Sequencing Chips. *Cell* **170,** 35–47.e13 (2017).

49. Sorek, R., Lawrence, C. M. & Wiedenheft, B. CRISPR-Mediated Adaptive Immune Systems in Bacteria and Archaea. *Annu. Rev. Biochem.* **82,** 237–266 (2013).

50. Wright, A. V., Nuñez, J. K. & Doudna, J. A. Biology and Applications of CRISPR Systems: Harnessing Nature's Toolbox for Genome Engineering. *Cell* **164,** 29–44 (2016).

51. Amitai, G. & Sorek, R. CRISPR-Cas adaptation: insights into the mechanism of action. *Nat. Rev. Microbiol.* **14,** 67–76 (2016).

52. Hsu, P. D., Lander, E. S. & Zhang, F. Development and Applications of CRISPR-Cas9 for Genome Engineering. *Cell* **157,** 1262–1278 (2014).

53. Sander, J. D. & Joung, J. K. CRISPR-Cas systems for editing, regulating and targeting genomes. *Nat. Biotechnol.* **32,** 347–355 (2014).

54. Crosetto, N. *et al.* Nucleotide-resolution DNA double-strand break mapping by next-generation sequencing. *Nat. Methods* **10,** 361–365 (2013).

55. Kim, D. *et al.* Digenome-seq: genome-wide profiling of CRISPR-Cas9 off-target effects in human cells. *Nat. Methods* **12,** 237–243, 1 p following 243 (2015).

56. Kuscu, C., Arslan, S., Singh, R., Thorpe, J. & Adli, M. Genome-wide analysis reveals characteristics of off-target sites bound by the Cas9 endonuclease. *Nat. Biotechnol.* **32,** 677–683 (2014).

57. O'Geen, H., Henry, I. M., Bhakta, M. S., Meckler, J. F. & Segal, D. J. A genome-wide analysis of Cas9 binding specificity using ChIP-seq and targeted sequence capture. *Nucleic Acids Res.* gkv137 (2015). doi:10.1093/nar/gkv137

58. Ran, F. A. *et al.* In vivo genome editing using Staphylococcus aureus Cas9. *Nature* **520,** 186–191 (2015).

59. Wu, X. *et al.* Genome-wide binding of the CRISPR endonuclease Cas9 in mammalian cells. *Nat. Biotechnol.* **32,** 670–676 (2014).

60. Fu, Y. *et al.* High-frequency off-target mutagenesis induced by CRISPR-Cas nucleases in human cells. *Nat. Biotechnol.* **31,** 822–826 (2013).

61. Wu, X., Kriz, A. J. & Sharp, P. A. Target specificity of the CRISPR-Cas9 system. *Quant. Biol.* **2,** 59–70 (2014).

62. Bolukbasi, M. F., Gupta, A. & Wolfe, S. A. Creating and evaluating accurate CRISPR-Cas9 scalpels for genomic surgery. *Nat. Methods* **13,** 41–50 (2016).

63. Nutiu, R. *et al.* Direct measurement of DNA affinity landscapes on a high-throughput sequencing instrument. *Nat. Biotechnol.* **29,** 659–664 (2011).

64. Buenrostro, J. D. *et al.* Quantitative analysis of RNA-protein interactions on a massively parallel array reveals biophysical and evolutionary landscapes. *Nat. Biotechnol.* **32,** 562–568 (2014).

65. Tome, J. M. *et al.* Comprehensive analysis of RNA-protein interactions by high-throughput sequencing-RNA affinity profiling. *Nat. Methods* **11,** 683–688 (2014).

66. Luo, M. L., Mullis, A. S., Leenay, R. T. & Beisel, C. L. Repurposing endogenous type I CRISPR-Cas systems for programmable gene repression. *Nucleic Acids Res.* gku971 (2014). doi:10.1093/nar/gku971

67. Makarova, K. S. *et al.* An updated evolutionary classification of CRISPR-Cas systems. *Nat. Rev. Microbiol.* **13,** 722–736 (2015).

68. Caliando, B. J. & Voigt, C. A. Targeted DNA degradation using a CRISPR device stably carried in the host genome. *Nat. Commun.* **6,** ncomms7989 (2015).

69. Leenay, R. T. *et al.* Identifying and Visualizing Functional PAM Diversity across CRISPR-Cas Systems. *Mol. Cell* **62,** 137–147 (2016).

70. Deveau, H. *et al.* Phage Response to CRISPR-Encoded Resistance in Streptococcus thermophilus. *J. Bacteriol.* **190,** 1390–1400 (2008).

71. Heler, R. *et al.* Cas9 specifies functional viral targets during CRISPR-Cas adaptation. *Nature* **519,** 199–202 (2015).

72. Horvath, P. *et al.* Diversity, Activity, and Evolution of CRISPR Loci in Streptococcus thermophilus. *J. Bacteriol.* **190,** 1401–1412 (2008).

73. Marraffini, L. A. & Sontheimer, E. J. CRISPR interference: RNA-directed adaptive immunity in bacteria and archaea. *Nat. Rev. Genet.* **11,** 181–190 (2010).

74. Mojica, F. J. M., Díez-Villaseñor, C., García-Martínez, J. & Almendros, C. Short motif sequences determine the targets of the prokaryotic CRISPR defence system. *Microbiology* **155,** 733–740 (2009).

75. Marraffini, L. A. CRISPR-Cas immunity in prokaryotes. *Nature* **526,** 55–61 (2015).

76. Semenova, E. *et al.* Interference by clustered regularly interspaced short palindromic repeat (CRISPR) RNA is governed by a seed sequence. *Proc. Natl. Acad. Sci.* **108,** 10098–10103 (2011).

77. Hayes, R. P. *et al.* Structural basis for promiscuous PAM recognition in type I–E Cascade from E. coli. *Nature* **advance online publication,** (2016).

78. Efron, B. & Tibshirani, R. J. *An Introduction to the Bootstrap*. (Chapman and Hall/CRC, 1993).

79. Guizar-Sicairos, M., Thurman, S. T. & Fienup, J. R. Efficient subpixel image registration algorithms. *Opt. Lett.* **33,** 156 (2008).

80. Zitová, B. & Flusser, J. Image registration methods: a survey. *Image Vis. Comput.* **21,** 977–1000 (2003).

81. Brown, L. G. A Survey of Image Registration Techniques. *ACM Comput Surv* **24,** 325–376 (1992).

82. Blosser, T. R. *et al.* Two distinct DNA binding modes guide dual roles of a CRISPR-Cas protein complex. *Mol. Cell* **58,** 60–70 (2015).

83. Redding, S. *et al.* Surveillance and Processing of Foreign DNA by the Escherichia coli CRISPR-Cas System. *Cell* **163,** 854–865 (2015).

84. Rutkauskas, M. *et al.* Directional R-Loop Formation by the CRISPR-Cas Surveillance Complex Cascade Provides Efficient Off-Target Site Rejection. *Cell Rep.* **10,** 1534–1543 (2015).

85. Szczelkun, M. D. *et al.* Direct observation of R-loop formation by single RNA-guided Cas9 and Cascade effector complexes. *Proc. Natl. Acad. Sci.* **111,** 9798–9803 (2014).

86. Carlson, C. D. *et al.* Specificity landscapes of DNA binding molecules elucidate biological function. *Proc. Natl. Acad. Sci.* **107,** 4544–4549 (2010).

87. Huo, Y. *et al.* Structures of CRISPR Cas3 offer mechanistic insights into Cascade-activated DNA unwinding and degradation. *Nat. Struct. Mol. Biol.* **21,** 771–777 (2014).

88. Jackson, R. N. *et al.* Crystal structure of the CRISPR RNA–guided surveillance complex from Escherichia coli. *Science* **345,** 1473–1479 (2014).

89. Mulepati, S., Héroux, A. & Bailey, S. Crystal structure of a CRISPR RNA–guided surveillance complex bound to a ssDNA target. *Science* **345,** 1479–1484 (2014).

90. Zhao, H. *et al.* Crystal structure of the RNA-guided immune surveillance Cascade complex in Escherichia coli. *Nature* **515,** 147–150 (2014).

91. van Erp, P. B. G. *et al.* Mechanism of CRISPR-RNA guided recognition of DNA targets in Escherichia coli. *Nucleic Acids Res.* **43,** 8381–8391 (2015).

92. Stormo, G. D. & Zhao, Y. Determining the specificity of protein-DNA interactions. *Nat. Rev. Genet.* **11,** 751–760 (2010).

93. Hsu, P. D. *et al.* DNA targeting specificity of RNA-guided Cas9 nucleases. *Nat. Biotechnol.* **31,** 827–832 (2013).

94. Fineran, P. C. *et al.* Degenerate target sites mediate rapid primed CRISPR adaptation. *Proc. Natl. Acad. Sci. U. S. A.* **111,** E1629-1638 (2014).

95. Wiedenheft, B. *et al.* RNA-guided complex from a bacterial immune system enhances target recognition through seed sequence interactions. *Proc. Natl. Acad. Sci. U. S. A.* **108,** 10092–10097 (2011).

96. Xue, C. *et al.* CRISPR interference and priming varies with individual spacer sequences. *Nucleic Acids Res.* **43,** 10831–10847 (2015).

97. Jore, M. M. *et al.* Structural basis for CRISPR RNA-guided DNA recognition by Cascade. *Nat. Struct. Mol. Biol.* **18,** 529–536 (2011).

98. Faehnle, C. R., Elkayam, E., Haase, A. D., Hannon, G. J. & Joshua-Tor, L. The making of a slicer: activation of human Argonaute-1. *Cell Rep.* **3,** 1901–1909 (2013).

99. Jiang, F., Zhou, K., Ma, L., Gressel, S. & Doudna, J. A. A Cas9–guide RNA complex preorganized for target DNA recognition. *Science* **348,** 1477–1481 (2015).

100. Nakanishi, K. *et al.* Eukaryote-specific insertion elements control human ARGONAUTE slicer activity. *Cell Rep.* **3,** 1893–1900 (2013).

101. Schirle, N. T. & MacRae, I. J. The crystal structure of human Argonaute2. *Science* **336,** 1037–1040 (2012).

102. Press, W. H. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. (Cambridge University Press, 2007).

103. Bertin, E. & Arnouts, S. SExtractor: Software for source extraction. *Astron. Astrophys. Suppl. Ser.* **117,** 12 (1996).

104. Maneewongvatana, S. & Mount, D. M. It's okay to be skinny, if your friends are fat. in *Center for Geometric Computing 4th Annual Workshop on Computational Geometry* **2,** 1–8 (1999).

105. Sashital, D. G., Wiedenheft, B. & Doudna, J. A. Mechanism of Foreign DNA Selection in a Bacterial Adaptive Immune System. *Mol. Cell* **46,** 606–615 (2012).

106. Semenova, E. *et al.* Highly efficient primed spacer acquisition from targets destroyed by the Escherichia coli type I-E CRISPR-Cas interfering complex. *Proc. Natl. Acad. Sci.* **113,** 7626–7631 (2016).

107. Staals, R. H. J. *et al.* Interference-driven spacer acquisition is dominant over naive and primed adaptation in a native CRISPR–Cas system. *Nat. Commun.* **7,** 12853 (2016).

108. Xue, C., Whitis, N. R. & Sashital, D. G. Conformational Control of Cascade Interference and Priming Activities in CRISPR Immunity. *Mol. Cell* **64,** 826–834 (2016).

109. Calisher, C. H., Childs, J. E., Field, H. E., Holmes, K. V. & Schountz, T. Bats: Important Reservoir Hosts of Emerging Viruses. *Clin. Microbiol. Rev.* **19,** 531–545 (2006).

110. Smith, I. & Wang, L.-F. Bats and their virome: an important source of emerging viruses capable of infecting humans. *Curr. Opin. Virol.* **3,** 84–91 (2013).

111. Francis, C. M. & Barrett, P. *A guide to the mammals of Southeast Asia*. (Princeton University Press Princeton, New Jersey, 2008).

112. Greenhall, A. M., Joermann, G. & Schmidt, U. Desmodus rotundus. *Mamm. Species Arch.* **202,** 1–6 (1983).

113. Riskin, D. K. & Hermanson, J. W. Biomechanics: Independent evolution of running in vampire bats. *Nature* **434,** 292–292 (2005).

114. Jones, K. E., Purvis, A., MacLARNON, A., Bininda-Emonds, O. R. P. & Simmons, N. B. A phylogenetic supertree of the bats (Mammalia: Chiroptera). *Biol. Rev.* **77,** 223–259 (2002).

115. Agnarsson, I., Zambrana-Torrelio, C. M., Flores-Saldana, N. P. & May-Collado, L. J. A time-calibrated species-level phylogeny of bats (Chiroptera, Mammalia). *PLoS Curr.* **3,** (2011).

116. Daugherty, M. D. & Malik, H. S. Rules of Engagement: Molecular Insights from Host-Virus Arms Races. *Annu. Rev. Genet.* **46,** 677–700 (2012).

117. Demogines, A., Abraham, J., Choe, H., Farzan, M. & Sawyer, S. L. Dual Host-Virus Arms Races Shape an Essential Housekeeping Protein. *PLOS Biol.* **11,** e1001571 (2013).

118. Kimura, M. Preponderance of synonymous changes as evidence for the neutral theory of molecular evolution. *Nature* **267,** 275–276 (1977).

119. Yang, Z. & Bielawski, J. P. Statistical methods for detecting molecular adaptation. *Trends Ecol. Evol.* **15,** 496–503 (2000).

120. Meyerson, N. R. & Sawyer, S. L. Two-stepping through time: mammals and viruses. *Trends Microbiol.* **19,** 286–294 (2011).

121. Ng, M. *et al.* Filovirus receptor NPC1 contributes to species-specific patterns of ebolavirus susceptibility in bats. *eLife* **4,**

122. Leroy, E. M. *et al.* Fruit bats as reservoirs of Ebola virus. *Nature* **438,** 575–576 (2005).

123. Towner, J. S. *et al.* Marburg Virus Infection Detected in a Common African Bat. *PLOS ONE* **2,** e764 (2007).

124. Towner, J. S. *et al.* Isolation of Genetically Diverse Marburg Viruses from Egyptian Fruit Bats. *PLOS Pathog.* **5,** e1000536 (2009).

125. Shaw, T. I. *et al.* Transcriptome Sequencing and Annotation for the Jamaican Fruit Bat (Artibeus jamaicensis). *PLoS ONE* **7,** e48472 (2012).

126. Dong, D., Lei, M., Liu, Y. & Zhang, S. Comparative inner ear transcriptome analysis between the Rickett's big-footed bats (Myotis ricketti) and the greater short-nosed fruit bats (Cynopterus sphinx). *BMC Genomics* **14,** 1 (2013).

127. Low, D. H. W. *et al.* Dracula's children: Molecular evolution of vampire bat venom. *J. Proteomics* **89,** 95–111 (2013).

128. Gracheva, E. O. *et al.* Ganglion-specific splicing of TRPV1 underlies infrared sensation in vampire bats. *Nature* **476,** 88–91 (2011).

129. Francischetti, I. M. B. *et al.* The "Vampirome": Transcriptome and proteome analysis of the principal and accessory submaxillary glands of the vampire bat Desmodus rotundus, a vector of human rabies. *J. Proteomics* **82,** 288–319 (2013).

130. Wang, Z. *et al.* Unique expression patterns of multiple key genes associated with the evolution of mammalian flight. *Proc. R. Soc. B Biol. Sci.* **281,** 20133133–20133133 (2014).

131. Fushan, A. A. *et al.* Gene expression defines natural changes in mammalian lifespan. *Aging Cell* **14,** 352–365 (2015).

132. Zhang, G. *et al.* Comparative Analysis of Bat Genomes Provides Insight into the Evolution of Flight and Immunity. *Science* **339,** 456–460 (2013).

133. Wu, L. *et al.* Deep RNA Sequencing Reveals Complex Transcriptional Landscape of a Bat Adenovirus. *J. Virol.* **87,** 503–511 (2013).

134. Phillips, C. J. *et al.* Dietary and Flight Energetic Adaptations in a Salivary Gland Transcriptome of an Insectivorous Bat. *PLoS ONE* **9,** e83512 (2014).

135. Papenfuss, A. T. *et al.* The immune gene repertoire of an important viral reservoir, the Australian black flying fox. *BMC Genomics* **13,** 261 (2012).

136. Lei, M., Dong, D., Mu, S., Pan, Y.-H. & Zhang, S. Comparison of Brain Transcriptome of the Greater Horseshoe Bats (Rhinolophus ferrumequinum) in Active and Torpid Episodes. *PLoS ONE* **9,** e107746 (2014).

137. Bolger, A. M., Lohse, M. & Usadel, B. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics* **30,** 2114–2120 (2014).

138. Grabherr, M. G. *et al.* Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nat. Biotechnol.* **29,** 644–652 (2011).

139. Robertson, G. *et al.* De novo assembly and analysis of RNA-seq data. *Nat. Methods* **7,** 909–912 (2010).

140. Martin, J. A. & Wang, Z. Next-generation transcriptome assembly. *Nat. Rev. Genet.* **12,** 671–682 (2011).

141. Altschul, S. F. *et al.* Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* **25,** 3389–3402 (1997).

142. Jarvis, E. D. *et al.* Whole-genome analyses resolve early branches in the tree of life of modern birds. *Science* **346,** 1320–1331 (2014).

143. Kent, J. kentUtils. GitHub Repository. (https://github.com/ENCODE-DCC/kentUtils) (2014).

144. Eddy, S. *HMMER User's Guide. Biological Sequence Analysis Using Profile Hidden Markov Models*. (2003).

145. Jones, K. E., Purvis, A., MacLARNON, A., Bininda-Emonds, O. R. P. & Simmons, N. B. A phylogenetic supertree of the bats (Mammalia: Chiroptera). *Biol. Rev.* **77,** 223–259 (2002).

146. Springer, M. S., Teeling, E. C., Madsen, O., Stanhope, M. J. & Jong, W. W. de. Integrated fossil and molecular data reconstruct bat echolocation. *Proc. Natl. Acad. Sci.* **98,** 6241–6246 (2001).

147. Hoofer, S. R., Reeder, S. A., Hansen, E. W. & Bussche, R. A. V. D. Molecular Phylogenetics and Taxonomic Review of Noctilionoid and Vespertilionoid Bats (Chiroptera: Yangochiroptera). *J. Mammal.* **84,** 809–821 (2003).

148. Teeling, E. C. *et al.* A Molecular Phylogeny for Bats Illuminates Biogeography and the Fossil Record. *Science* **307,** 580–584 (2005).

149. Tsagkogeorga, G., Parker, J., Stupka, E., Cotton, J. A. & Rossiter, S. J. Phylogenomic Analyses Elucidate the Evolutionary Relationships of Bats. *Curr. Biol.* **23,** 2262–2267 (2013).

150. Hoofer, S. R. & Bussche, R. A. V. D. Molecular Phylogenetics of the Chiropteran Family Vespertilionidae. *Acta Chiropterologica* **5,** 1–63 (2003).

151. IUCN. Murina leucogaster: Stubbe, M., Ariunbold, J., Buuveibaatar, V., Dorjderem, S., Monkhzul, T., Otgonbaatar, M., Tsogbadrakh, M., Francis, C.M., Bates, P.J.J. & Csorba, G.: The IUCN Red List of Threatened Species 2016: e.T13943A22093328. (2016). doi:10.2305/IUCN.UK.2016-2.RLTS.T13943A22093328.en

152. IUCN. Myotis pilosus: Csorba, G. & Bates, P.: The IUCN Red List of Threatened Species 2008: e.T14193A4418772. (2008). doi:10.2305/IUCN.UK.2008.RLTS.T14193A4418772.en

153. IUCN. Myotis davidii: Smith, A.T., Johnston, C.H., Jones, G. & Rossiter, S.: The IUCN Red List of Threatened Species 2008: e.T136250A4265409. (2008). doi:10.2305/IUCN.UK.2008.RLTS.T136250A4265409.en

154. IUCN. Myotis brandtii: Hutson, A.M., Spitzenberger, F., Coroiu, I., Aulagnier, S., Juste, J., Karataş, A., Palmeirim, J. & Paunović, M.: The IUCN Red List of Threatened Species 2008: e.T14125A4397500. (2008). doi:10.2305/IUCN.UK.2008.RLTS.T14125A4397500.en

155. IUCN. Myotis lucifugus: Arroyo-Cabrales, J. & Álvarez-Castañeda, S.T.: The IUCN Red List of Threatened Species 2008: e.T14176A4415629. (2008). doi:10.2305/IUCN.UK.2008.RLTS.T14176A4415629.en

156. Wetterer, A. L., Rockman, M. V. & Simmons, N. B. Phylogeny of phyllostomid bats (mammalia: chiroptera): data from diverse morphological systems, sex chromosomes, and restriction sites. *Bull. Am. Mus. Nat. Hist.* 1–200 (2000). doi:10.1206/0003-0090(2000)248<0001:POPBMC>2.0.CO;2

157. Rojas, D., Warsi, O. M. & Dávalos, L. M. Bats (Chiroptera: Noctilionoidea) Challenge a Recent Origin of Extant Neotropical Diversity. *Syst. Biol.* **65,** 432–448 (2016).

158. Yang, Z. PAML 4: Phylogenetic Analysis by Maximum Likelihood. *Mol. Biol. Evol.* **24,** 1586–1591 (2007).

159. Wright, R. M., Aglyamova, G. V., Meyer, E. & Matz, M. V. Gene expression associated with white syndromes in a reef building coral, Acropora hyacinthus. *BMC Genomics* **16,** 371 (2015).

160. Francis, W. R. *et al.* A comparison across non-model animals suggests an optimal sequencing depth for de novo transcriptome assembly. *BMC Genomics* **14,** 167 (2013).

161. Harris, R. S. Improved pairwise alignment of genomic DNA. (The Pennsylvania State University, 2007).

162. Katoh, K. & Standley, D. M. MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability. *Mol. Biol. Evol.* **30,** 772–780 (2013).

163. Suyama, M., Torrents, D. & Bork, P. PAL2NAL: robust conversion of protein sequence alignments into the corresponding codon alignments. *Nucleic Acids Res.* **34,** W609–W612 (2006).

164. Ashburner, M. *et al.* Gene Ontology: tool for the unification of biology. *Nature Genetics* (2000). doi:10.1038/75556

165. Expansion of the Gene Ontology knowledgebase and resources. *Nucleic Acids Res.* **45,** D331–D338 (2017).

166. Edelstein, A. D. *et al.* Advanced methods of microscope control using μManager software. *J. Biol. Methods* **1,** 10 (2014).