Bioinformatics, 36, 2020, i560–i568
doi: 10.1093/bioinformatics/btaa505
ISMB 2020

CRISPRLAND: Interpretable large-scale inference of DNA repair landscape based on a spectral approach

Amirali Aghazadeh, Orhan Ocal and Kannan Ramchandran*

Department of Electrical Engineering and Computer Science, University of California, Berkeley, Berkeley 94720, CA, USA

Corresponding email ID:kannanr@eecs.berkeley.edu
*To whom correspondence should be addressed.

Abstract

Summary: We propose a new spectral framework for reliable training, scalable inference and interpretable explanation of the DNA repair outcome following a Cas9 cutting. Our framework, dubbed CRISPRLAND, relies on an unexploited observation about the nature of the repair process: the landscape of the DNA repair is highly sparse in the (Walsh–Hadamard) spectral domain. This observation enables our framework to address key shortcomings that limit the interpretability and scaling of current deep-learning-based DNA repair models. In particular, CRISPRLAND reduces the time to compute the full DNA repair landscape from a striking 5230 years to 1 week and the sampling complexity from 10^{12} to 3 million guide RNAs with only a small loss in accuracy ($R^2R^2 \sim 0.9$). Our proposed framework is based on a divide-and-conquer strategy that uses a fast *peeling* algorithm to learn the DNA repair models. CRISPRLAND captures lower-degree features around the cut site, which enrich for short insertions and deletions as well as higher-degree microhomology patterns that enrich for longer deletions.

Availability and implementation: The CRISPRLAND software is publicly available at https://github.com/UCBASiCS/CRISPRLand.

Contact: kannanr@eecs.berkeley.edu

1 Introduction

Recent studies on site-specific double-stranded breaks (DSBs) generated by the RNA-guided DNA endonuclease Cas9 have shown that the DNA repair outcome following Cas9 cutting is non-random and consistent across experimental replicates, cell lines and reagent delivery methods and highly a function of the DNA sequence around the cut site (van Overbeek *et al.*, 2016). Follow-up studies have emerged which employ machine-learning models, such as deep neural networks (Shen *et al.*, 2018), decision trees (Leenay *et al.*, 2019) and logistic regression (Allen *et al.*, 2019) to train DNA repair models using a database of DSBs generated in CRISPR-Cas9 experiments. These models take in the DNA sequence of the genome in a small window surrounding the cut and predict key statistics of the repair outcome, such as the percentage of cells with in-frame shifts, insertions and deletions (Fig. 1A).

While initial studies have used the DNA repair models to design gene knock-out experiments in therapeutically important cell types, such as T cells (Leenay *et al.*, 2019), as well as gene knock-in experiments, to edit certain genes (Shen *et al.*, 2018), the applicability of the DNA repair models is severely hindered in a large-scale setting due to the lack of a reliable, interpretable and scalable machinery to train and analyze the DNA repair models. Here, we discuss these shortcomings in detail:

Reliability. Currently, there is no clear mechanism to determine the number of site-specific CRISPR experiments required for the reliable training of DNA repair models. The number of experiments is typically defined by the total budget invested in the experiments rather than a principled scientific mechanism backed by theoretical or computational reasoning. In addition, the location of cut sites (i.e. the guide RNAs) on the genome are determined either by random selection or enforced by other studies rather than experimental design procedures tailored to the DNA repair models.

Interpretability. Our understanding of how the current DNA repair models operate is extremely limited; e.g. we do not have a clear mechanism (other than ad hoc interpretation methods for general purpose deep neural networks) to determine what features or combination of features enrich for key repair outcomes, such as in-frame shifts, insertions and deletions. A powerful ineterpretable model would enable a mechanistic understanding of the repair process as well as a set of design rules for gene knock-out and knock-in experiments.

Scalability. Finally, the DNA repair models are extremely slow in the inference time. Running these models against the cut sites in the potential coding region on the human genome takes months on a regular computer. Considering the growing interest in gene editing and knock-out experiments in various cell types in the resolutions of single cells, there is a critical need for more scalable methods at the inference time.

Here, we aim to address these problems by a new computational framework, dubbed CRISPRLAND, which analyzes the landscape of the DNA repair process in the spectral domain. CRISPRLAND focuses on the microhomology-mediated end-joining (MMEJ) and non-homologous end-joining (NHEJ) (Sonoda *et al.*, 2006) repair processes for which machine-learning models have been recently developed. The key insight of our framework is that the MMEJ and NHEJ repair processes can be modeled by a pseudo-Boolean function $x[\mathbf{m}] = x(m_1, m_2, \ldots, m_n)$: $\mathbb{F}_2^n \rightarrow \mathbb{R}$, where m_i are binary variables that encode the input DNA sequence of length ℓ surrounding the cut site, $\mathbb{F}_2$ refers to finite field consisting $\{0, 1\}$, and $x[\mathbf{m}]$ is a

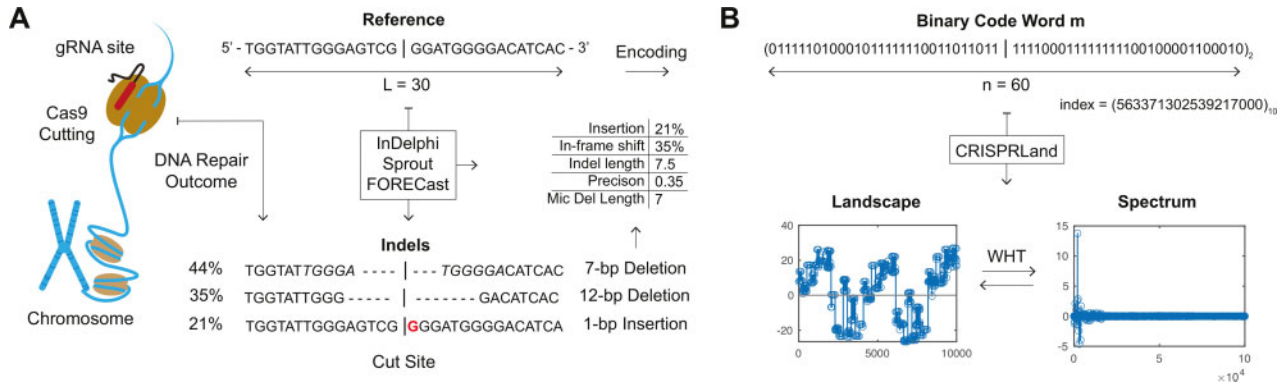

Fig. 1. Schematic of the CRISPRLAND framework. (**A**) A Cas9 induced DSB is illustrated on the genome with the reference sequence surrounding the cut site as well as the three indels resulting from the DNA repair with their corresponding frequencies predicted by the machine-learning models [inDelphi (Shen *et al.*, 2018), Sprout (Leenay *et al.*, 2019) and FORECast (Allen *et al.*, 2019)]. The 7-bp deletion indicates a microhomology-mediated deletion. The repeating patterns surrounding the deletion have been italicized for better visualization. Five key DNA repair outcomes including fraction of cells with an insertion, fraction of cells with an in-frame shift, average length of an indel, precision and average length of the microhomology-mediate deletions have been calculated. (**B**) The binary representation of the reference sequence is illustrated and its corresponding decimal index in the CRISPRLAND framework. The DNA repair landscape (in terms of the percentage of insertions) as well as the spectrum of the landscape is illustrated for 10 000 ordered gRNA sequences evaluated using CRISPRLAND

real-valued outcome of interest, e.g. percentage of insertions (Fig. 1B).

CRISPRLAND's approach in analyzing the repair model $x[\mathbf{m}]$ stems from a key theorem (Boros and Hammer, 2002) in mathematics, which states that any pseudo-Boolean function $f(z_1, z_2, \ldots, z_n)$ can be represented uniquely by a multi-linear polynomial over the hyper cube $(z_1, z_2, \ldots, z_n) \in \{-1, +1\}^n$:

$$f(z_1, z_2, \ldots, z_n) = \sum_{\mathcal{S} \subseteq [n]} \alpha_{\mathcal{S}} \prod_{i \in \mathcal{S}} z_i, \tag{1}$$

where $\mathcal{S}$ is a subset of $\{1, 2, 3, \ldots, n\} = [n]$ and $\alpha_{\mathcal{S}}$ is the Walsh–Hadamard transform (WHT) coefficient associated with the monomial $\prod_{i \in \mathcal{S}} z_i$. For example, the pseudo-Boolean function

$$f(z_1, z_2, z_3, z_4, z_5) = 12 z_1 z_4 - 3 z_3 + 6 z_1 z_2 z_5, \tag{2}$$

has three monomials with degrees 2, 1 and 3 and WHT coefficients 12, –3 and 6, respectively. Note that, the pseudo-Boolean $f(.)$ in this example is considered to be sparse since out of $2^5 = 32$ possible monomials only three of them are active (i.e. have non-zero coefficient).

If we replace z_i with $(-1)^{m_i}$ such that z_i=1 when m_i=0 and $z_i = -1$ when m_i=1, we have $x[\mathbf{m}] = f((-1)^{m_1}, (-1)^{m_2}, \ldots, (-1)^{m_n})$ for $\mathbf{m} \in \mathbb{F}_2^n$ and the WHT $X[\mathbf{k}] = \sqrt{n}\alpha_{\mathcal{S}}$ such that $\text{supp}(\mathbf{k}) = \mathcal{S}$, where supp(.) indicates the non-zeros coordinates (called the support function in signal processing). The non-zero WHT coefficients reveal higher order interactions between the input features that enrich (or deplete) certain repair outcomes. Our main goal is to efficiently estimate these sparse coefficients from the underlying biological model and provide the biological interpretations behind them.

To this end, we first formally define the notion of the fitness landscape (also called landscape in this article) of the DNA repair process using the pseudo–Boolean function $x[\mathbf{m}]$ and then show the steps to develop CRISPRLAND in order to compute the landscape based on the recently developed DNA repair models. We, then analyze the repair landscape in the WHT spectral domain. We observe that the WHT coefficients $\alpha_{\mathcal{S}}$ of the fitness landscape are surprisingly sparse and the level of sparsity depends on the type of the repair outcome. In particular, the landscape in terms of the insertion percentage is sparser than that of the in-frame shift percentage and the rest of the repair outcomes.

1.1 Contributions

The sparsity of the DNA repair outcome in the WHT domain enables CRISPRLAND to address key shortcomings regarding the training, inference, and interpretation of the current repair models. First, CRISPRLAND provides a method to significantly reduce the number of site-specific gRNAs to fully recover the DNA repair landscape. The idea is to leverage the sparsity of the landscape in the WHT spectral domain. Similar analysis has revolutionized sensing systems in areas, such as medical imaging, radio astronomy and radar/sonar imaging. Second, CRISPRLAND provides a concrete framework to explain the black-box machine-learning models by simply inspecting the non-zero coefficients of the WHT. In particular, it reveals patterns that are indicative of deletions that are mediated by microhomologies, i.e. repeating patterns around the cut site. These patterns are highly difficult to observe using classical interpretation tools in machine learning since they reflect higher order interactions between the features. From this perspective, CRISPRLAND serves as a complimentary tool-set to explain deep-learning-based DNA repair models. Third, the new massive-scale computational algorithm developed in CRISPRLAND recovers the complete DNA repair landscape exponentially faster compared to the conventional way of repeatedly querying the machine-learning model. This speedup is achieved by employing a divide-and-conquer strategy building on concepts from signal processing and coding theory. Our algorithm allows us to recover a model for the DNA repair outcome through a fast *peeling* algorithm based on a number of carefully designed input guide RNAs whose cardinality scales *logarithmically* with the size of the landscape. We show how such design scheme enables CRISPRLAND to extrapolate the repair landscape in massive scales; a task that is practically impossible using computers today. CRISPRLAND reduces the computational complexity to compute the full repair landscape from a striking 5230 years to 1 week and the sampling complexity from 10^{12} to 3 million guide RNAs. Our findings are demonstrated using the state-of-the-art models trained on recent experimental datasets from DNA repair in Cas9-mediated DSBs.

2 Repair landscape

We will consider two landscapes of scientific interest, which symbolize two DNA repair outcomes with maximum differences in the WHT spectral domain. One is the percentage of cells with an insertions and the other one is the fraction of cells with an in-frame shift. Other DNA repair outcomes, such as deletion percentage, average indel length, average deletion length and precision can be analyzed similarly. We will instead provide a summary of those landscapes in terms of WHT complexity.

2.1 Insertion percentage

In order to compute the fitness landscape of the DNA repair outcome, we first generate all the different 4^{ℓ} possible binary code

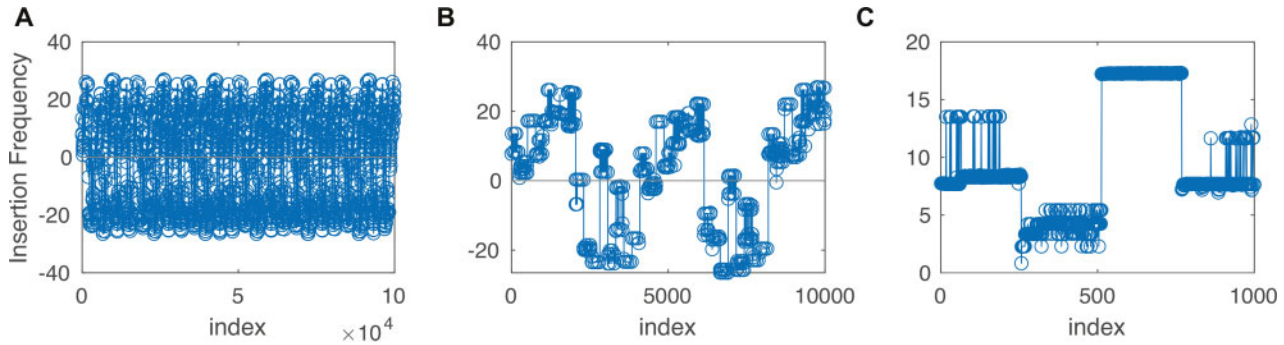

Fig. 2. The fitness landscape of DNA repair outcome in U2OS in terms of the insertion percentage. Only the first (**A**) 100 000, (**B**) 10 000 and (**C**) 1000 coordinates of the landscape are illustrated for more clarity. Insertion percentages are mean-subtracted. The landscape clearly shows redundant structures in different resolutions

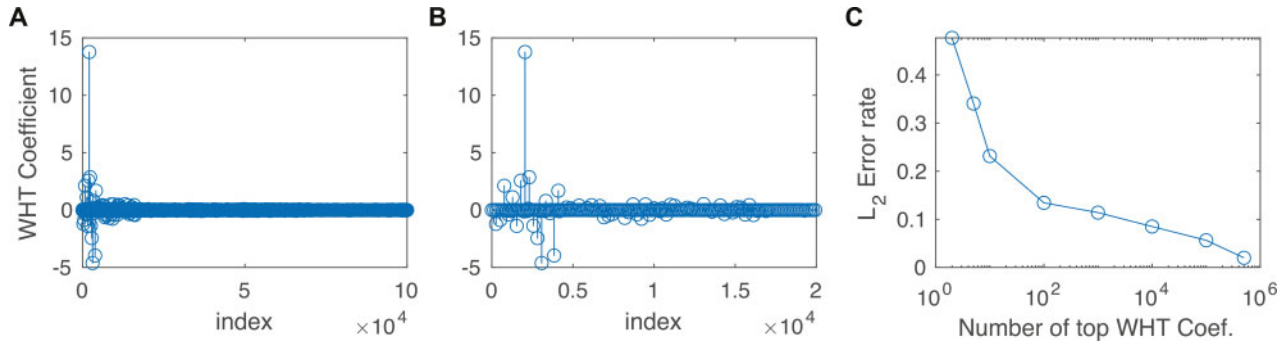

Fig. 3. The WHT of the repair landscape for insertion percentage in U2OS. Only the first (**A**) 100 000 and (**B**) 20 000 coordinates of the landscape are illustrated for more clarity. (**C**) The ℓ_2 error in recovering the landscape using the top coefficients of the WHT transform is illustrated in U2OS

words $\mathbf{m}$ of length $n = 2\ell$ ordered in a way that two adjacent binary codes are different only by one bit. We then encode the binary code words into their corresponding DNA sequences. While various encoding strategies can be used to encode the DNA sequence into a binary code; in this article, we use the following encoding: A : 00, T : 01, C : 10 and G : 11. With this encoding, we can represent a DNA sequence of length ℓ using a binary code $\mathbf{m}$ of length $n = 2\ell$. Other encoding will result in a similar analysis.

We then construct CRISPR experiments where we introduce a DSB on the genome at a point exactly in the middle of the DNA sequence constructed above. We repeat the same experiment for all the 4^ℓ generated DNA sequences and use an already trained repair model [e.g. inDelphi (Shen *et al.*, 2018), Sprout (Leenay *et al.*, 2019) or FORECast (Allen *et al.*, 2019)] to find the repair outcome for all the generated DNA sequences. The outcomes can be obtained experimentally; however, here, we use a trained repair model as a proof-of-concept. We compute and store the key summary statistics (outputs) of the DNA repair outcome including, precision, in-frame shift percentage, insertion percentage, deletion percentage and indel length in 4^ℓ-dimensional vectors in the same order as they appear in the binary code $\mathbf{m}$; we call these vectors the fitness landscape of the DNA repair outcome. The fitness landscape fully describes the DNA repair models in terms of the outputs mentioned above. Note that, the repair landscape inherits the real-world statistics of the repair data, such as the presence of the protospacer adjacent motif around the cut site, since the original machine-learning models are trained on real-world experimental data.

An example of the fitness landscape in terms of the insertion percentage is illustrated in Figure 2 for $\ell = 10$ in human bone Osteosarcoma epithelial cells (U2OS). The figure illustrates the mean-subtracted insertion percentage of the first 100 000, 10 000 and 1000 coordinates of the $4^{10} = 1\,048\,576$-dimensional landscape. In order to maintain the minimum input sequence length requirement of the DNA repair outcome prediction models, we append the DNA sequences from left and right with two fixed short DNA sequences of length 15. In the next section, we elaborate on the computational challenges in increasing ℓ to larger values and we develop a new scheme that enables us to increase ℓ to arbitrarily larger values and exploring the fitness landscape fully.

We applied the WHT on the resulting 4^{10}-dimensional landscape signal. The result is depicted in Figure 3 in different resolutions. In order to measure how sparse the WHT transform is, we also plot the recovery error of the inverse WHT using only the top most components of the WHT in terms of the ℓ_2 norm. Our key observation is that the WHT transform of the landscape is surprisingly sparse; only the top-100 coefficients out of the total of 4^{10} coefficients (<0.01%) suffices to recover the landscape with around 10% error. The sparsity in the WHT domain is a consequence of the nature of the repair process; it would not have appeared in a random landscape, which would have a fully dense (i.e. flat) WHT spectrum. Our finding in Figure 3 also suggests a fundamental bound in terms of the number of samples required to approximate the function $x[\mathbf{m}]$. This suggests the minimum number of CRISPR experiments required for the reliable training of the DNA repair models.

Recent studies have shown that the DNA repair outcome is a function of the cell type. In other words, if we break the genome at the very same locations but in two different cell types we get two different repair outcomes. An intriguing question is how such variations across cell type affect the coefficients in the spectral domain. We compared the WHT of the fitness landscape of repair models across three cell types: U2OS, mESC (mouse embryonic stem cell) and HCT116 (human colon cancer cells) in Figure 4. The index of the top WHT coefficients stays surprisingly consistent across the cell types, while the values change. Our analysis on the landscape of the other repair outcomes (not shown here), such as in-frame shift percentage, indel length and precision, also shows the consistency across the set of top WHT coefficients. Further studies are required to test and explore this hypothesis; however, our results seem to suggest a shared repair dynamic across cell types with small mechanistic differences.

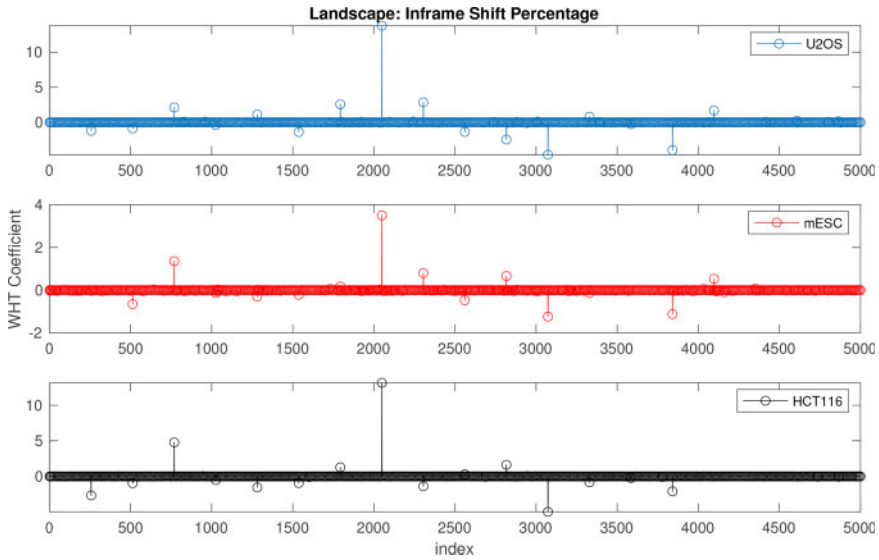

Fig. 4. The first 5000 coefficients of WHT of the repair landscape in terms of the percentage of insertion in three cell types: U2OS, mESC and HCT116. The index of large WHT coefficients is consistent across cell types

Table 1. Top-5 WHT coefficients of the repair landscape for percentage of insertions in U2OS cells

Index	Code	Monomial	Interpretation	Coefficient
2048	00 00 00 00 10 | 00 00 00 00 00	z_9	{C, G}^{A, T}@−1?	13.76
3072	00 00 00 00 11 | 00 00 00 00 00	z_9z_{10}	*−1 bp?*	−4.61
3840	00 00 00 00 11 | 11 00 00 00 00	$z_9z_{10}z_{11}z_{12}$	*−1 & +1 bp?*	−3.96
2304	00 00 00 00 10 | 01 00 00 00 00	z_9z_{12}	{C, G}^{A, T}@−1 & {T, G}^{A, C}@+1?	2.84
1792	00 00 00 00 01 | 11 00 00 00 00	$z_{10}z_{11}z_{12}$	*+1 bp?* {T, G}^{A, C}@−1?	2.56

Note: WHT coefficients correspond to monomials defined by their binary expansions. Each monomial can be easily interpreted as a question regarding the type of the nucleotides around the cut site.

Knowing that the landscape of the repair outcome can be recovered using only a small fraction of the WHT coefficients, a natural question is what biological features the top WHT coefficients corresponds to. The theorem in Equation (1) suggests that each WHT coefficient α_S corresponds to a monomial defined by $\mathcal{S}$. Table 1 tabulates the index of the top-5 WHT coefficients, their corresponding binary representation, monomial and a short interpretation of each monomial. Each code in the second column is the binary representation of the index written in $n = 2\ell = 20$ binary digits. The monomial is obtained based on the pattern of '1's in the binary code. The repair model $x[\mathbf{m}]$ can be written in terms of the first five monomials as,

$$x[\mathbf{m}] \approx 13.76z_9 - 4.61z_9z_{10} - 3.96z_9z_{10}z_{11}z_{12} + 2.84z_9z_{12} + 2.56z_{10}z_{11}z_{12}, \quad (3)$$

where $z_i = (-1)^{m_i}$. Note that, the cut site is in between z_{10} and z_{11} and is indicated by a straight line in Table 1, and the sign of the WHT coefficients indicates the direction of influence.

In order to better understand the interpretation of each monomial, the biological meaning of the first monomial z_9 will be described in details, and the rest follows from the same intuition. The monomial $z_9 = (-1)^{m_9}$ only activates when the ninth binary digit in one. Let us recall our encoding policy mentioned earlier: A : 00, T : 01, C : 10 and G : 11. Based on this encoding, the monomial z_9 only activates when the nucleotide next to the cut site from the 5' end (we call it location –1) is either a C or a G nucleotide since these are the nucleotides that their binary code end in the digit 1. Therefore, this monomial is only asking if the nucleotide at location –1 is C/G or A/T. We represent this question using the logical statement $\{C, G\}^\wedge\{A, T\}@-1?$ where $\wedge$ is an OR operator and @ points to a location on the genome. This feature has also shown to be have a significant correlation with the percentage of insertions (Leenay *et al.*, 2019; Shen *et al.*, 2018). The rest of the monomials can be interpreted similarly.

All the top-five monomials showing up in the WHT of the fitness landscape for the insertion percentage are related to the nucleotides that are adjacent to the cut site. This shows that the nucleotides around the cut are sufficient to fully describe the DNA repair landscape for the insertion percentage. As the next section shows, this locality around the cut might not carry over to other landscapes.

2.2 In-frame shift percentage

We conduct a similar landscape analysis for another important repair outcome: the percentage of cells with in-frame shifts. Predicting the percentage of in-frame shift is another critical problem in designing efficient gene knock-out experiments. The WHT of the repair landscape for in-frame shifts is illustrated in Figure 5. The non-zero coordinates (i.e. the support) in the WHT domain is more wide spread and consists of coefficients that are periodically repeated along the spectrum. The spectrum is denser than the insertion percentage. In particular, more than 10 000 WHT coefficients out of the total of 4^{10} coefficients (>1%) is required to recover the landscape with around 10% error. This suggests that the information required to estimate the in-frame shift percentage are more widespread and has higher frequency components compared to insertion percentage, which is more localized to the nucleotides around the cut site.

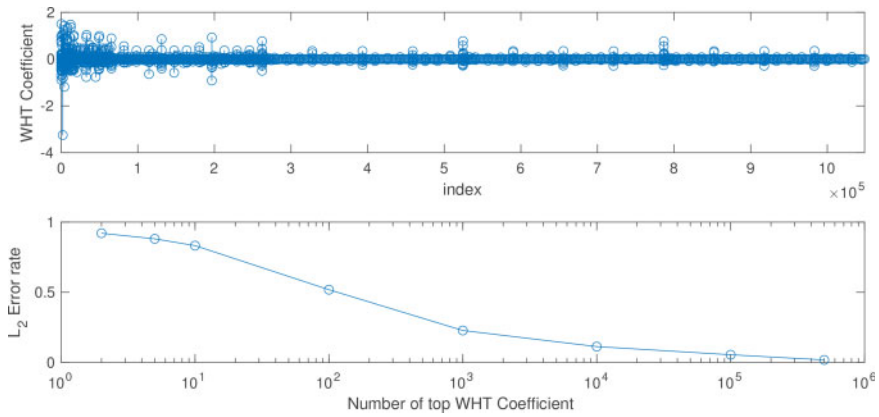

Fig. 5. (**A**) The WHT of the repair landscape for in-frame percentage is illustrated in U2OS. (**B**) The WHT of in-frame percentage is denser than that of the insertion percentage. This suggests that more number of samples is required to recover the landscape of in-frame percentage compared to insertion percentage

Table 2. Top-5 WHT coefficients of the CRISPR fitness landscape for the in-frame shift percentage in U2OS cells

Index	Code	Monomial	Interpretation	Coefficient
2048	00 00 00 00 10 | 00 00 00 00 00	z_9	*{C, G}^{A, T}@−1?*	−3.24
256	00 00 00 00 00 | 01 00 00 00 00	z_{12}	*{T, G}^{A, C}@+1?*	1.50
12 288	00 00 00 11 00 | 00 00 00 00 00	z_7z_8	*−2 bp?*	1.48
4160	00 00 00 01 00 | 00 01 00 00 00	$^*z_8z_{14}$	*{T, G}^{A, C}@-2 & +2 bp?*	1.42
12 480	00 00 00 11 00 | 00 11 00 00 00	$^*z_7z_8z_{13}z_{14}$	*−2 & +2 bp?*	1.36

Note: WHT coefficients correspond to monomials defined by their binary expansions. Each monomial can be easily interpreted as a question regarding the type of the nucleotides around the cut site. The monomials with * sign indicate a microhomology feature, which are identical patterns repeating around the cut site and enrich for a deletion outcome.

The top-five WHT coefficients α_S of the repair landscape are tabulated in Table 2. Similar to the landscape of insertion percentage, it can be seen that the first two monomials correspond to nucleotides that are just next to the cut site. The last two monomials, however, show a new pattern. These two monomials ask about symmetric patterns that occur around the cut site. These features resemble the microhomology patterns, i.e. repeating sequences around the cut site. Presence of a microhomology pattern is known to enrich for a deletion outcome (Sonoda *et al.*, 2006). We will show the effect of macrologies in large-scale experiments in the next section.

3 Massive-scale landscape extrapolation

We now discuss the computational challenges in finding the full landscape of the repair outcome when the context sequence length goes beyond $\ell > 10$ and describe an algorithm that uses ideas from coding theory and signal processing in order to scale to these large dimensions. Note that, the number of possible DNA sequences of length ℓ grows exponentially with ℓ as 4^ℓ. Testing a single DNA sequence using the inDelphi software (Shen *et al.*, 2018) takes about 0.15 s on a 2.7 GHz Intel Core i5 with 8 G of RAM. Therefore, it takes around $4^{10} \times 0.15 = 157\,286$ s (close to 2 days) to obtain the full landscape of the repair outcome of each cell type *in silico* with $\ell = 10$. However, running the same inference problem for a longer sequence of length $\ell = 20$ (i.e. just twice the length of the previous experiment) takes about 5220 years. This volume of computation cannot be done on the computers today. Needless to say that, even with today; s multiplexing technologies, doing as many experiments is also completely out of picture.

Before talking about our massive-scale algorithm to handle larger values of ℓ, we want motivate that it is in fact necessary to consider larger values of ℓ to accurately estimate the landscape of DNA repair outcome. We determine what is the minimum value of ℓ to be considered to accurately estimate the landscape. To this end, we perform a systemic analysis of DNA repair outcomes as a function of the window size around the cut site. The aim is to find out what values of ℓ enables us to capture most of the variation in the repair outcomes.

A window is considered around the cut size with varying length. The DNA sequence in the window is varied and the maximum range that the DNA repair outcome changes is monitored as a function of the window size (nucleotide on each side of the cut) and plotted in Figure 6. While knowing only the 3 nt around the cut site (window of length 6) reduces the range of variation for insertion percentage below 5%, we need to know at least 20 nt (a window of size 40) to reduce the range of variation for other outcomes below 5%. This raises the need to develop a computational platform that scales to such large values of the window size.

While it might seem that we need $4^{\ell=20}$ number of experiments to obtain the full landscape, the actual number of samples required is much smaller. This is achieved through exploiting the structure of the landscape, i.e. the landscape of DNA repair outcome is sparse in the WHT domain. If we approximate the WHT of the landscape using only K number of non-zeros elements (by keeping only the top-K coefficients), results from compressed sensing (Baraniuk, 2007; Donoho, 2006) show that only $\mathcal{O}(K\log(4^\ell))$ (i.e. $\mathcal{O}(K\ell)$) number of random samples are sufficient to recover the landscape. However, achieving an algorithm whose *computational complexity* also scales gracefully as the dimensions of the problem grows is challenging. This is the algorithmic challenge that we tackle in CRISPRLAND using a divide-and-conquer strategy. The description of the CRISPRLAND algorithm is provided in Section 4.

We demonstrate that CRISPRLAND approximates the DNA repair landscape in terms of the insertion percentage and in-frame shifts in U2OS cells with a context window of size $\ell = 20$ using only about 3 million carefully designed gRNAs (samples) from coding theory. CRISPRLAND designs the input samples using a variant of the SPRIGHT algorithm (Li *et al.*, 2015) as we will describe in

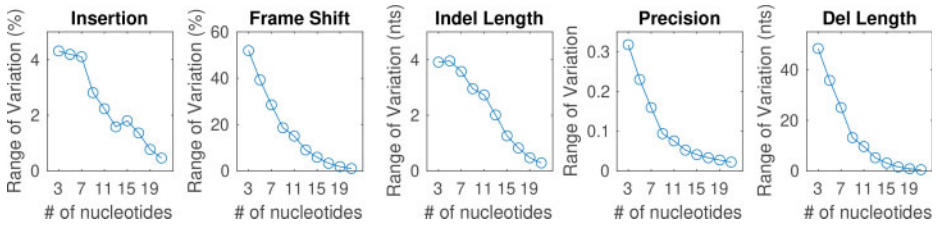

Fig. 6. The variation of the DNA repair landscape as a function of the window size around the cut site (number of nucleotides from each side of the cut). For all the outcomes, except for the Insertion Percentage, a window of 20 nt from each side is required to minimize the variation and fully capture the landscape

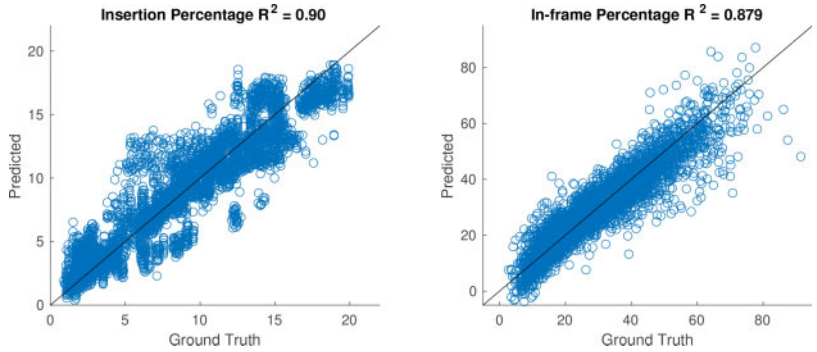

Fig. 7. The prediction results of CRISPRLAND on 330 000 randomly chosen unseen test CRISPR experiments. The analysis has been done on the sparse insertion percentage landscape as well as the less sparse in-frame shift percentage landscape. Only 10 000 data points are plotted for more clarity. In both landscapes, CRISPRLAND predicts the outcomes with very high accuracy

Section 4. Note that, in general, the DNA sequences of all the gRNAs suggested by CRISPRLAND might not exactly appear on the human genome. In those cases, either a sufficiently close gRNA can be selected or the recovery algorithm can be changed to accommodate for the deviation from the designed gRNAs. Obviously, this will not be a problem when the repair outcomes are derived using an already trained repair model.

As we will discuss in the next section, CRISPRLAND uses a modified version of SPRIGHT's recovery algorithm to find the full landscape. Here, we evaluate the generalization performance of CRISPRLAND using 330 000 *unseen* samples from the DNA repair landscape. The results of our prediction algorithm can be seen in Figure 7. CRISPRLAND predicts the DNA repair outcome of the set of 330 000 unseen test gRNAs with a very high accuracy ($R^2 \sim 0.9$) in both the easier insertion percentage and the harder in-frame shift percentage landscapes. Given the repair outcome of the 3 million designed gRNAs (which takes about a week to acquire from inDelphi's software), CRISPRLAND requires only few seconds to recover the repair outcome of the queried gRNAs.

4 Materials and methods

Our CRISPRLAND algorithm for learning the landscape for the DNA repair outcome is based on computing a sparse WHT. WHT is analogous to Fourier Transform for functions that take Boolean variables. More precisely, let $N = 2^n$ for a non-negative integer n, and let $x \in \mathbb{R}^N$ be a vector. We can index the elements of x with an n-length binary sequence $m \in \mathbb{F}_2^n$. WHT of x is then defined as follows

$$X_k = \frac{1}{\sqrt{N}} \sum_{m \in \mathbb{F}_2^n} (-1)^{\langle k,m \rangle} x_m, \tag{4}$$

where $\langle k, m \rangle = \sum_{i=0}^{n-1} k_i m_i$ with the addition operation being over $\mathbb{F}_2$. Using the coefficients X_k, one can recover the vector x as

$$x_m = \frac{1}{\sqrt{N}} \sum_{k \in \mathbb{F}_2^n} (-1)^{\langle m,k \rangle} X_k. \tag{5}$$

WHT can be viewed as recovering a multinomial representation of a function. Consider a pseudo-Boolean function $f : \{-1, 1\}^n \to \mathbb{R}$ that takes n variables with values in $\{-1, 1\}$ and outputs a real number. Every such function has a unique expansion of the form

$$f(v_0, \ldots, v_{n-1}) = \frac{1}{\sqrt{N}} \sum_{k \in \mathbb{F}_2^n} X_k \prod_{i:k_i=1} v_i. \tag{6}$$

Let, x_m be the evaluation of this polynomial at $v_i = (-1)^{m_i}$. Then the kth Walsh–Hadamard coefficient X_k corresponds to the coefficient of the multinomial term $\prod_{i:k_i=1} v_i$. Hence, WHT can be seen as recovering the coefficients of multinomial expansion of a pseudo-Boolean function.

The problem we are interested in is recovering the WHT coefficients (equivalently, the pseudo-Boolean function) when there is sparsity in the WHT domain. Methods proposed in compressed sensing literature can be used to recover a sparse signal in a sample efficient way (Donoho, 2006). However, the algorithms proposed in the literature like OMP (Tropp, 2004) or LASSO (Tibshirani, 1996) requires operations that scale at least linearly with the ambient dimension N. On the other hand, our method requires sublinear computational complexity whenever the degrees of freedom K scale sublinearly with the ambient dimension N (Li *et al.*, 2015). The key properties of the CRISPRLAND algorithm are presented in the following theorem.

Theorem 1 [(Li *et al.*, 2015)] *Let $\alpha \in (0, 1)$ be a fixed number. Suppose $N = 2^n$ and assume $K = N^\alpha$. Let $x \in \mathbb{R}^N$ be a vector and $X \in \mathbb{R}^N$ be its WHT. Assume that X is K-sparse and support is selected uniformly at random among all possible $\binom{n}{k}$ subsets of $[n]$ of size K. Then, there is an algorithm with the following properties:*

1. *Sample complexity: algorithm uses $\mathcal{O}(K \log^2 N)$ samples of x.*
2. *Computational complexity: total number of operations to successfully decode all non-zero WHT coefficients or declare a decoding failure is $\mathcal{O}(K \log^3 N)$.*

3. *Success probability: probability of recovering X completely approaches 1 as N grows, where the probability is taken over randomness of selecting the support of X.*

This speedup is achieved by employing a divide-and-conquer strategy, where we break the problem of recovering a K-sparse signal into K-many smaller problems of recovering 1-sparse signal, and solve each 1-sparse problem efficiently, and combine the solutions to each of them to recover the original signal. The recovery algorithm is closely tied to decoding a sparse-graph-code through *peeling*, and we use techniques from the literature on low-density parity check (LDPC) codes (Richardson and Urbanke, 2008) and product codes (Elias, 1954) for deriving our method.

Note that, under the assumptions of the theorem, theoretically, order of $K\log(N)$ samples is required for learning the correct model by information theoretic arguments (Li *et al.*, 2015). The algorithm described here, which requires $K\log^2(N)$ samples is off from order optimality by only a logarithmic factor. As a matter of fact, the algorithm can be tweaked to be order optimal (Li *et al.*, 2015). However, that version of the algorithm is not described in this article as it requires a complex additional step.

The first step of the algorithm is to generate linear mixing of transform domain coefficients based on the following property.

Property 1 *Let x be an $N=2^n$ length vector. Given a shift vector $p\in\mathbb{F}^n$ and a full-rank subsampling matrix $\mathbf{H}\in\mathbb{F}_2^{b\times n}$, let y be the vector of length $B=2^b$, where $y_m=x_{m\mathbf{H}+p}$ for all $m\in\mathbb{F}_2^b$. Then, the WHT coefficients of y satisfy*

$$Y_k=\sqrt{\frac{B}{N}}\sum_{j\in\mathbb{F}_2^N:j\mathbf{H}^\top=k}(-1)^{\langle p,j\rangle}X_j, \tag{7}$$

where X_j is the jth WHT coefficient of x.

The above property states that the WHT coefficients X_k are modulated by $(-1)^{\langle p,k\rangle}$ when a shift of p is applied to the indices of x, and that subsampling of the input signal creates a linear mixing of WHT coefficients.

Using Property 1, we create linear mixing of coefficients by choosing C many subsampling matrices $\mathbf{H}_1,\ldots,\mathbf{H}_C$, where each matrix is $b\times n$ dimensional. Furthermore, we choose for each subsampling $\mathbf{P}_1,\ldots,\mathbf{P}_C$ shift matrices, where each of them is $\mathcal{O}(\log^2 N)\times n$ dimensional. The choice of C, the matrices $\mathbf{H}_i$ and the delays $\mathbf{P}_i$ for $i=1,\ldots,C$ are going to be described in the following sections. Then WHT coefficients are calculated for the shifted-and-subsampled sequences. We give an example below for the linear mixing resulting from subsampling.

Example 1 *Let x be a vector of length 16, and let us define $y^{(1)}_m=2x_{\mathbf{H}_1 m}$ and $y^{(2)}_m=2x_{\mathbf{H}_2 m}$, where*

$$\mathbf{H}_1=\begin{pmatrix}0&0\\0&0\\1&0\\0&1\end{pmatrix},\quad \mathbf{H}_2=\begin{pmatrix}1&0\\0&1\\0&0\\0&0\end{pmatrix}.$$

From property 1, we see that all the WHT coefficients of x whose binary index have the same last two digits is hashed to the same bin (underlined in the following equations) for $y^{(1)}$, i.e. we have

$$\begin{aligned}
Y^{(1)}_{00}&=X_{0\underline{000}}+X_{0\underline{100}}+X_{1\underline{000}}+X_{1\underline{100}},\\
Y^{(1)}_{01}&=X_{0\underline{001}}+X_{0\underline{101}}+X_{1\underline{001}}+X_{1\underline{101}},\\
Y^{(1)}_{10}&=X_{0\underline{010}}+X_{0\underline{110}}+X_{1\underline{010}}+X_{1\underline{110}},\\
Y^{(1)}_{11}&=X_{0\underline{011}}+X_{0\underline{111}}+X_{1\underline{011}}+X_{1\underline{111}}.
\end{aligned}$$

Similarly, for $y^{(2)}$ we get

$$\begin{aligned}
Y^{(2)}_{00}&=X_{\underline{0000}}+X_{\underline{0001}}+X_{\underline{0010}}+X_{\underline{0011}},\\
Y^{(2)}_{01}&=X_{\underline{0100}}+X_{\underline{0101}}+X_{\underline{0110}}+X_{\underline{0111}},\\
Y^{(2)}_{10}&=X_{\underline{1000}}+X_{\underline{1001}}+X_{\underline{1010}}+X_{\underline{1011}},\\
Y^{(2)}_{11}&=X_{\underline{1100}}+X_{\underline{1101}}+X_{\underline{1110}}+X_{\underline{1111}}.
\end{aligned}$$

Under the assumptions of Theorem 1 on sparsity and the support of the non-zero WHT coefficients of the signal, the linear mixing of coefficients take a form where they can be solved for through *peeling*. The following provides an example of such linear mixing.

Example 2 *Let $x\in\mathbb{R}^{16}$ have WHT coefficients equal to*

$$X_k=\begin{cases}X_{0001} & \text{if k}=0001,\\ X_{0100} & \text{if k}=0100,\\ X_{0101} & \text{if k}=0101,\\ X_{1010} & \text{if k}=1010,\\ 0 & \text{otherwise.}\end{cases}$$

Under the subsampling used in example 1 the WHT coefficients of the sub-sampled vectors satisfy

$$\begin{aligned}
Y^{(1)}_{00}&=X_{0\underline{100}}, & Y^{(2)}_{00}&=X_{\underline{0001}},\\
Y^{(1)}_{01}&=X_{0\underline{001}}+X_{0\underline{101}}, & Y^{(2)}_{01}&=X_{\underline{0100}}+X_{\underline{0101}},\\
Y^{(1)}_{10}&=X_{1\underline{010}}, & Y^{(2)}_{10}&=X_{\underline{1010}},\\
Y^{(1)}_{11}&=0, & Y^{(2)}_{11}&=0.
\end{aligned}$$

We give the details of peeling algorithm in reference to this example in the following section.

4.1 Recovery through peeling with an oracle

The relationship between the measurements and the unknown coefficients can be shown as a bipartite graph. The graph related to the linear mixing in Example 2 and the recovery of the non-zero coefficients is illustrated in Figure 8. The unknown coefficients are shown on the left and referred to as *variable nodes*, and the measurements are shown on the right and referred to as *check nodes*. An edge is drawn between a variable node and a check node if the unknown coefficient related to that variable node contributes to the measurement related to that check node. Each check node can be categorized into the following three types:

1. Zero-ton: a check node is a zero-ton if it has no non-zero coefficients (shaded in white in Fig. 8).
2. Single-ton: a check node is a single-ton if it involves only one non-zero coefficient (shaded in blue in Fig. 8). Specifically, we refer to the index k and its associated value X_k as the index-value pair (k,X_k).
3. Multi-ton: a check node is a multi-ton if it contains more than one non-zero coefficient (shaded in orange in Fig. 8).

To illustrate the peeling algorithm for recovery, we assume that there exists an 'oracle' that informs the decoder exactly which check nodes are single-tons, and provides the index-value pair for that single-ton. In Example 2, in the first round of peeling (shown in Fig. 8A), the oracle informs the decoder that the check nodes corresponding to $Y^{(1)}_{00}$, $Y^{(1)}_{10}$, $Y^{(2)}_{00}$ and $Y^{(2)}_{10}$ are single-tons with index-value pairs $(0100,X_{0100})$, $(1010,X_{1010})$, $(0001,X_{0001})$ and $(1010,X_{1010})$, respectively. Then the decoder can subtract their contributions from other check nodes, forming new single-tons.

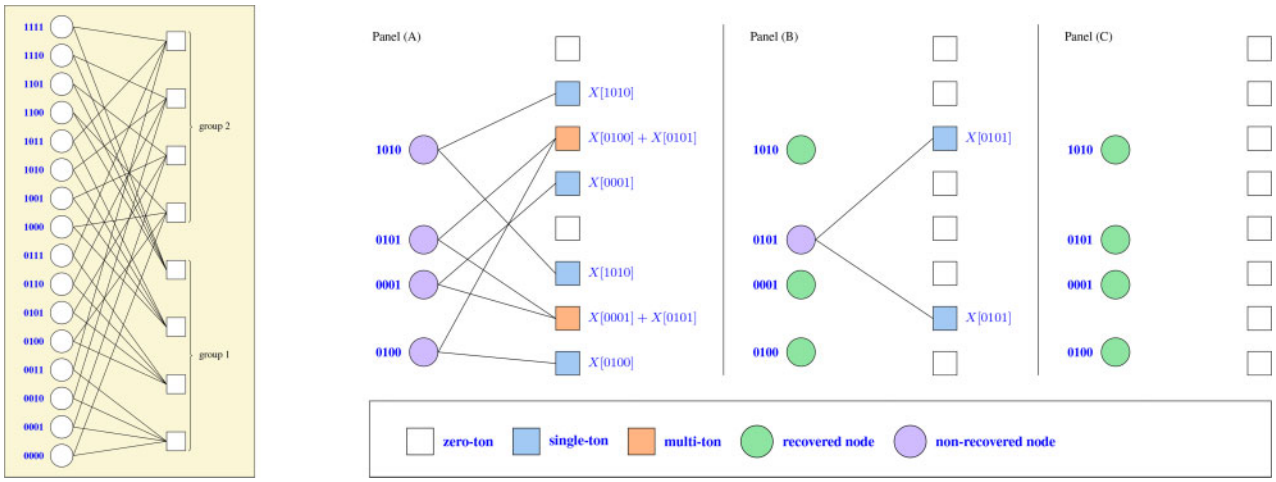

Fig. 8. (Left) The connections between the variable nodes (WHT coefficients) and the check nodes (measurements) in Example 1. (Right) Recovering the unknown coefficients in Example 2. The graph induced by the non-zero coefficients is shown in panel (**A**). In the first round of peeling, we recover coefficients at indices 0100, 001 and 1010, and get the graph in panel (**B**). In two rounds of peeling, all the non-zero elements of the signal are recovered as shown in panel (**C**)

Therefore, with the oracle information, the peeling decoder repeats the following steps:

1. select all the edges in the bipartite graph with right degree 1 (identify single-ton bins);
2. remove (peel off) these edges as well as the corresponding pair of variable and check nodes connected to these edges;
3. remove (peel off) all other edges connected to the variable nodes that have been removed in Step 2;
4. subtract the contributions of the variable nodes from the check nodes whose edges have been removed in Step 3.

Decoding is successful if all the edges are removed from the graph.

In this work, we choose the subsampling matrices uniformly at random over $F^{b\times n}$. Other constructions alongside with their theoretical guarantees can be found at Li *et al.* (2015) and Scheibler *et al.* (2015). We chose the random design as it is observed to have superior practical performance in some regimes of interest (Ocal *et al.*, 2019; Scheibler *et al.*, 2015).

Since the proof of the algorithm follows the same steps as in Li *et al.* (2015), we just provide a sketch here and refer the interested readers to that article. Since the sparsity is uniformly distributed, each non-zero entry of X is connected to a check node chosen uniformly at random in each subsampling group. This results in a left-regular LDPC code construction, and the proof for recovering the support X follows the same steps in Li *et al.* (2015).

In peeling, we recover a variable node (non-zero coefficient of X) if it is connected to a check node with degree 1, and remove the outgoing edges from that variable node. The *density evolution* is a powerful tool in modern coding theory that tracks the average density of remaining edges in the graph after ℓ rounds of peeling (Richardson and Urbanke, 2008). The density evolution equations for our setting is given by the recursive equation

$$p_\ell = (1 - e^{-d_{p\ell-1}/(M/K)})^{d-1}, \tag{8}$$

where $p_0 = 1$, and M is the total number of parity check nodes. This assumes that the depth ℓ neighborhood of the chosen edge is a tree. We can show similarly to Li *et al.* (2015) that the depth ℓ neighborhood of a randomly chosen edge is a tree with high probability for any fixed ℓ. On average, an arbitrarily large fraction of edges are removed if p_ℓ goes to zero as $\ell \to \infty$. For p_ℓ to go to zero, M/K needs to be greater than a threshold for a fixed d. These thresholds are shown in Table 3. Then, one can use the standard Doob's martingale argument to show that the fraction of non-recovered components concentrates around its mean (Richardson and Urbanke, 2001).

Table 3. Thresholds for recovery (Li *et al.*, 2015)

Groups	3	4	5	6
M/K	1.2218	1.2949	1.4250	1.5697

Note: M, number of check nodes; K, number of variable nodes (sparsity).

This guarantees recovery of arbitrarily large fraction of significant components. Then, an expander-graph argument is used to show that peeling continues until all of the coefficients are recovered (Li *et al.*, 2015).

4.2 Replacing the oracle

We now show how to replace the *oracle* in the peeling algorithm with a realizable mechanism. This is done by employing $\mathcal{O}(\log^2 N)$ shifts for each subsampling matrix where $\log(N)$ shifts are to recover each digit of the location k, and we take $\mathcal{O}(\log N)$ samples for each location for noise averaging. Let $U_{\mathbf{H},p}(k)$ be the kth WHT coefficient of the signal obtained by shifting indices of x by p and then subsampling by $\mathbf{H}$. From Property 1, we have

$$U_{\mathbf{H},p}(k) := \sqrt{\frac{B}{N}} \sum_{j:\mathbf{H}^\top = k} (-1)^{\langle j,p\rangle} X_j. \tag{9}$$

Furthermore, let us define the ratio of a WHT coefficient obtained by using the same subsampling matrix but using two different shifts

$$r_{A,p,q}(k) := \frac{U_{A,p+q}(k)}{U_{A,p}(k)}. \tag{10}$$

Assume that for a WHT index k in Equation (9), there is only one index j such that $A^\top j = k$ and $X_j \neq 0$ (i.e. the check node corresponding to it is a single-ton). Then, it follows that $U_{A,p}(k) = \sqrt{\frac{B}{N}}(-1)^{\langle j,p\rangle} X_j$. Using $q = e_i \in F^n$ (the vector with all indices =0 except for the ith index, which is =1) in Equation (10) yields

$$r_{A,p,e_i}(k) = \frac{(-1)^{\langle j,p+e_i\rangle} X_j}{(-1)^{\langle j,p\rangle} X_j} = (-1)^{\langle j,e_i\rangle}. \tag{11}$$

Note that, this value is in $\{-1,+1\}$ for all p if there is no noise. As the value of $\langle j, e_i\rangle$ is equal to the ith index of the location $j \in \mathbb{F}_2^n$, by using shifts $\{e_i\}_{i=0}^{n-1}$ going through all indices of j, we can recover

it. When there is noise, it can be shown that by taking $\mathcal{O}(\log N)$ random shifts, the probability of detecting the location wrongly can be made polynomially small (Li *et al.*, 2015).

5 Conclusion and discussion

While much of the body of research in training, testing and interpreting image and text models generalize to machine-learning problems in computational biology, there are yet distinct aspects in modern biological datasets that require new perspectives and methodologies. In several of the molecular biology and genomics datasets, as an example, the data points comprise long sequences of discrete elements, such as oligonucleotides (Alipanahi *et al.*, 2015), amino acids (Huang *et al.*, 2016) or mutations (Yang *et al.*, 2019) that can be represented using binary codes. These data modalities typically lack the common structures (e.g. invariance, etc.) present in image or text data while exhibiting other intriguing features.

In this article, we took an important problem in molecular biology, i.e. the problem of predicting the DNA repair outcome, as a model problem and demonstrated how the discreteness of the input data as well as the structure of the biological process can be exploited in our CRISPRLAND framework to do efficient training, inference and model interpretation using the ideas from coding theory and signal processing. The key observation that enables our analysis is the sparsity of the DNA repair models in the WHT spectral domain, which stems from the fact that the output of the model is a function of at most sparse number of monomials that capture distinct patterns around the cut site. Note that, both our results as well as the results in the DNA repair models employed in our work (e.g. InDelphi) explain more than 75% of the variance in the experimental data. The remaining unexplained variance due to other covariates, such as chromatin factor [see Leenay *et al.* (2019)], only slightly changes the repair landscape.

We demonstrate that CRISPRLAND reduces the time required to find the DNA repair landscape from thousands of years to couples of days using a logarithmically less number of site-specific Cas9 cuttings on the human genome and thus meets the burgeoning demand for large-scale CRISPR gene editing studies. Nevertheless, depending on the scale of the inference task, the window size around the cut site can be set to adjust the number of gRNAs to achieve a desired accuracy even in smaller-scale CRISPR experiments.

We speculate that several of the current machine-learning models trained for problems in computational biology would be sparse in WHT domain as well. The sparsity can be similarly exploited in these applications in various aspects including experimental design, interpretation and fast inference.

Funding

This work was supported by the National Science Foundation [CCF-Medium-1702678].

Conflict of Interest: none declared.

References

Alipanahi,B. *et al.* (2015) Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.*, **33**, 831–838.

Allen,F. *et al.* (2019) Predicting the mutations generated by repair of Cas9-induced double-strand breaks. *Nat. Biotechnol.*, **37**, 64–72.

Baraniuk,R.G. (2007) Compressive sensing. *IEEE Signal Process. Mag.*, **24**, 118–121.

Boros,E. and Hammer,P.L. (2002) Pseudo-Boolean optimization. *Discrete Appl. Math.*, **123**, 155–225.

Donoho,D.L. (2006) Compressed sensing. *IEEE Trans. Inf. Theory*, **52**, 1289–1306.

Elias,P. (1954) Error-free coding. *Trans. IRE Prof. Group Inf. Theory*, **4**, 29–37.

Huang,P.S. *et al.* (2016) The coming of age of de novo protein design. *Nature*, **537**, 320–327.

Leenay,R.T. *et al.* (2019) Large dataset enables prediction of repair after CRISPR–Cas9 editing in primary T cells. *Nat. Biotechnol.*, **37**, 1034–1037.

Li,X. *et al.* (2015) Spright: a fast and robust framework for sparse Walsh-Hadamard transform. *arXiv preprint arXiv: 1508.06336*.

Ocal,O. *et al.* (2019) Low-degree pseudo-Boolean function recovery using codes. In: *2019 IEEE International Symposium on Information Theory (ISIT)*. pp. 1207–1211. IEEE.

Richardson,T. and Urbanke,R. (2008) *Modern Coding Theory*. Cambridge University Press, Cambridge.

Richardson,T.J. and Urbanke,R.L. (2001) The capacity of low-density parity--check codes under message-passing decoding. *IEEE Trans. Inf. Theory*, **47**, 599–618.

Scheibler,R. *et al.* (2015) A fast Hadamard transform for signals with sublinear sparsity in the transform domain. *IEEE Trans. Inf. Theory*, **61**, 2115–2132.

Shen,M.W. *et al.* (2018) Predictable and precise template-free CRISPR editing of pathogenic variants. *Nature*, **563**, 646–651.

Sonoda,E. *et al.* (2006) Differential usage of non-homologous end-joining and homologous recombination in double strand break repair. *DNA Repair*, **5**, 1021–1029.

Tibshirani,R. (1996) Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Series B Methodol.*, **58**, 267–288.

Tropp,J.A. (2004) Greed is good: algorithmic results for sparse approximation. *IEEE Trans. Inf. Theory*, **50**, 2231–2242.

van Overbeek,M. *et al.* (2016) DNA repair profiling reveals nonrandom outcomes at Cas9-mediated breaks. *Mol. Cell*, **63**, 633–646.

Yang,K.K. *et al.* (2019) Machine-learning-guided directed evolution for protein engineering. *Nat. Methods*, **16**, 687–694.