A Content-Addressable DNA Database with Learned Sequence Encodings

Kendall Stewart[1], Yuan-Jyue Chen[2], David Ward[1], Xiaomeng Liu[1],
Georg Seelig[1], Karin Strauss[1,2], and Luis Ceze[1]

[1] Paul G. Allen School of Computer Science & Engineering, University of Washington
[2] Microsoft Research, Redmond, Washington

Abstract. We present strand and codeword design schemes for a DNA database capable of approximate similarity search over a multidimensional dataset of content-rich media. Our strand designs address crosstalk in associative DNA databases, and we demonstrate a novel method for learning DNA sequence encodings from data, applying it to a dataset of tens of thousands of images. We test our design in the wetlab using one hundred target images and ten query images, and show that our database is capable of performing similarity-based enrichment: on average, visually similar images account for 30% of the sequencing reads for each query, despite making up only 10% of the database.

1 Introduction

DNA-based databases were first proposed over twenty years ago by Baum [3], yet recent demonstrations of their practicality [4, 6, 8, 9, 18, 28] have generated a renewed interest into researching related theory and applications.

Some of these recent demonstrations of DNA storage have used key-based random access for their retrieval schemes, falling short of the content-based associative searches envisioned by Baum. Our goal is to close this gap and design a DNA-based digital data store equipped with a mechanism for content-based similarity search.

This work contributes two advances to the field of DNA storage: first, a strand design optimized for associative search. Second, a sequence encoder capable of preserving similarity between documents, such that a query sequence generated from a given document will retrieve similar documents from the database. We validate our designs with wetlab experiments.

While our methods should generalize to databases comprising any type of media, we focus on images in this work, as there is a rich body of prior work in content-based image retrieval to draw on.

The rest of this paper is laid out as follows: Section 2 covers background on similarity search, DNA-based parallel search, and DNA-based data storage. Section 3 details our strand designs. Section 4 describes our methodology for mapping images to DNA sequences. Section 5 outlines our experimental protocol and the results of our experiments. Section 6 discusses the results and proposes future work. Section 7 addresses related work, and Section 8 concludes the paper.

2 Background

2.1 Similarity Search

The problem of *similarity search* is to retrieve documents from a database that are similar in *content* to a given query. For media such as text, images and video, this can be a difficult task. Most state-of-the-art systems convert each document into a vector-space representation using either a hand-crafted embedding, or one learned via a neural network. These *feature vectors* can then be compared with metrics like Euclidean distance, where similar documents will tend to be close together in feature-space. Therefore, a similarity search can be reduced to a k-nearest-neighbor or R-near-neighbor search.

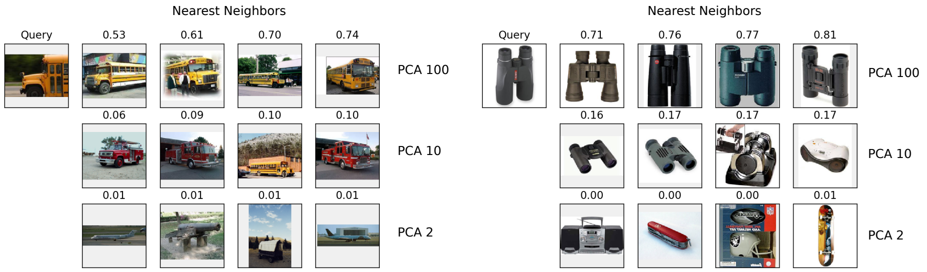


Fig. 1: A pair of sample queries from the Caltech-256 dataset, showing the four nearest neighbors in three different feature spaces. Each neighbor is annotated with its Euclidean distance to the query in that space.

Feature vectors that are effective for similarity search tend to be high dimensional. To illustrate this, Figure 1 shows two queries using the Caltech-256 image dataset [10]. The visual features of each image in the dataset were extracted using VGG16, a publicly available convolutional neural network trained on an image classification task. We used the 4096-dimensional activations from the FC2 layer, an intermediate layer in VGG16 whose activations have shown to be effective in content-based image retrieval tasks [25]. These features were reduced down to 100, 10, and 2 dimensions using principal component analysis (PCA). The nearest neighbors in each of these subspaces (with respect to Euclidean distance) are shown to the right of each query. Qualitatively, the nearest neighbors higher-dimensional spaces appear more similar to the query than the nearest neighbors in lower-dimensional spaces.

When feature vectors have hundreds of dimensions, the well-known “curse of dimensionality” defeats efficient indexing schemes [12]. In the worst case, every item in the database must be examined to find all images within a certain distance threshold. Relaxations of the search problem that allow for errors or omissions result in much faster lookups, using algorithms such as locality-sensitive hashing (LSH) [2].

Looking toward a future where zettabytes of data are generated every year [11], even techniques such as LSH that reduce the amount of data that needs to be inspected by orders of magnitude will still burden traditional storage with a tremendous number of IO requests to a massive storage infrastructure, outstripping the time and energy cost of the feature vector distance computation itself.

Computer architects have noticed that the power required to move data from the storage device to the compute unit can be reduced by moving the compute substrate closer to the storage substrate. This class of techniques is broadly called "near-data" processing [14].

2.2 DNA-based Parallel Search

"Adleman-style" DNA computing [1] can be thought of as an extreme version of near-data processing: each DNA strand is designed to both store and process information — the compute and storage substrates are the same.

Like Adleman's original solution to the Hamiltonian Path problem, this style of parallel processing requires exponential amounts of DNA to solve combinatorial problems. However, for less computationally intense problems like similarity search, the amount of DNA required is much less: if each of N items in the database is mapped to a single "target" molecule, then N identical copies of a "query" molecule are sufficient to react with every item in the database. If the query is equipped with a biotin tail and designed to hybridize only with relevant data, then relevant items can be "fished out" of the database using streptavidin-coated magnetic beads.

This amounts to an extremely high-bandwidth parallel search, in the vein of near-data processing techniques. Furthermore, because PCR can make exponentially many copies of the query molecule, the amount of DNA that needs to be directly synthesized is minimal. This makes DNA-based search especially appealing in the zettabyte-yottabyte future.

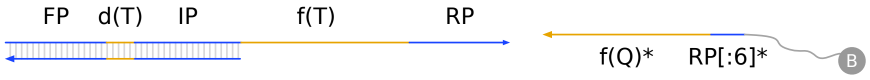
2.3 DNA-based Data Storage

The current state-of-the-art DNA storage systems (Organick et al. [18] includes a good survey of recent work) focus on zero-bit-error retrieval of arbitrary digital data. Each digital file is segmented and encoded into many thousands of unique sequences, and individual files can be retrieved from a mixed database using PCR-based random access. In this work, we focus on a database for storing and retrieving *metadata*. Instead of storing sequences that contain the complete file, each file is associated with a sequence that contains the semantic features used for content-based retrieval, as well as a pointer to the file in another database (which could be either a traditional database or a DNA-based one).

3 Database Design

To take advantage of the near-data processing capabilities of DNA, we need a database design that allows each element in the database to both store and process data. We choose to separate these two concerns by associating each database element with two sequences: one that stores an ID unique to that datum, and one that is generated from the semantic features of that datum, designed as a locus for a hybridization probe. The ID is not an "active" site, but rather the information to be retrieved by the search — for instance, it could be the address of the datum in another database that stores the document's complete data.

The simplest way to retain the association between the ID sequence and the feature sequence in a DNA database is to place them on the same strand of DNA. However, this association can cause unwanted secondary structures on longer strands, and can result in cross-talk if a query reacts with a potential target's ID sequence instead of its feature sequence.



(a) Database Element Design

(b) Query Design

Fig. 2: Strand Designs. Blue indicates a conserved region, orange indicates a region specific to that data item. Arrow indicates the 3' end. Star (*) indicates reverse complement. RP[:6] indicates the first six bases of domain RP.

Our strand designs address this issue, and are shown in Figure 2. The database entries (Figure 2a) are synthesized single-stranded, but are made partially double stranded using a single-step PCR reaction starting from IP (the "internal primer"), which is conserved across all elements in the database.

This process covers up the IP region, the ID sequence associated with the data ($d(T)$), and the forward primer (FP) region, which is another conserved region used to prepare samples for sequencing. This leaves the feature sequence ($f(T)$) and the conserved reverse sequencing primer (RP) available to interact with the query.

To execute a query Q, a biotinylated query strand (Figure 2b) is mixed with the prepared targets. Because the query and target feature sequences are designed to be imperfect matches, the query strand also includes the reverse complement of first six bases of RP (denoted RP[:6]*) — this exact match is designed to prevent misalignments and ensure that hybridization only depends on the interaction between $f(T)$ and $f(Q)$. The query and targets are annealed, and then streptavidin-coated magnetic beads are added to pull down targets that have hybridized with the queries.

The resulting filtered targets are amplified using FP and RP, then sequenced to retrieve the data region associated with each target.

4 Learned Sequence Encodings

To take advantage of the strand designs described above, we need to design a mapping from images to feature domains such that a query molecule will retrieve relevant targets from the database. To simplify our task, we pre-process all images by transforming them into the 10-dimensional subspace shown in Figure 1, and choose our feature domains to be 30 nucleotides in length.

Our general feature encoding strategy is inspired by semantic hashing [21], where a deep neural network transforms an input feature space into an output address space where similar items are "close" together. Our goal is to design a neural network sequence encoder that takes the 10-dimensional image feature vectors from the VGG16+PCA extraction process described in Section 2.1, and outputs DNA sequences that are close together if and only if the feature vectors are close together. Following Tsaftaris et al. [22], we define a pair of query and target sequences as "close" if their hybridization reaction has a high thermodynamic yield: the proportion of target molecules that are converted into a query-target duplex.

To train the neural network, we want a loss function that will push the encoder's parameters to generate output sequences where a query retrieves a target if and only if the target and query represent similar images. The most appropriate choice for this is the cross-entropy loss[3], where the labels are binary similarity labels (similar vs. not similar) for each pair of query and target images, and the retrieval probabilities are the thermodynamic yields of each query-target hybridization reaction.

Using the cross-entropy loss requires us to define a binary notion of image similarity, and to define thermodynamic yield as a differentiable function of two DNA sequences. The function must be differentiable because neural networks are efficiently trained using gradient descent, which requires taking the derivative of the loss with respect to the encoder parameters.

In the sections below, we present a definition of binary image similarity, followed by an approximation for thermodynamic yield using Hamming distance, and an approximation for Hamming distance using the cosine distance between "one-hot" encodings of DNA bases. Finally, we present the results of using these approximations to train a neural network on a large image dataset.

4.1 Binary Image Similarity

As described in Section 2.1, a semantic notion of image "similarity" can be mapped to a real-valued number by computing the Euclidean distance between two image feature vectors. However, to use the efficient cross-entropy loss function defined above, we must label image pairs with a binary label: "similar" or "not similar". The simplest way to do this is to apply a threshold to the Euclidean distance.

[3] Given a set of n pairs of binary labels $y \in \{0, 1\}$ and retrieval probabilities p, the cross-entropy loss is:

$$l(y, p) = -\frac{1}{n} \sum_{i=1}^{n} y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)$$

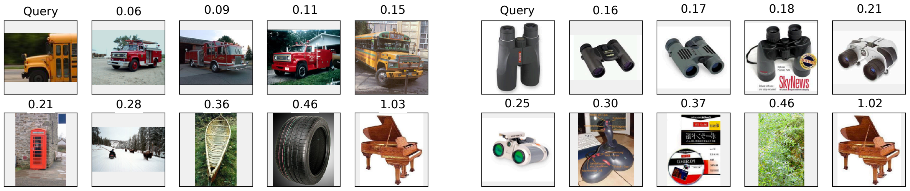


Fig. 3: Sample queries demonstrating the relationship between image similarity and distance in the 10-dimensional PCA subspace shown in Figure 1. Distances less than 0.2 usually correspond to similar images, while those greater than 0.2 do not.

Because the definition of similarity is ultimately up to a human observer, we must determine this threshold by inspection. For the feature extraction method we used, we found a threshold of 0.2 to be fairly reliable across the Caltech-256 dataset. Figure 3 demonstrates this for a pair of sample queries.

4.2 Approximating Thermodynamic Yield

Thermodynamic yield can be calculated accurately by using the multi-stranded partition function [5], which is used by tools such as NUPACK [29]. Unfortunately, this calculation is expensive and not differentiable, and thus cannot be used directly to train a neural network.

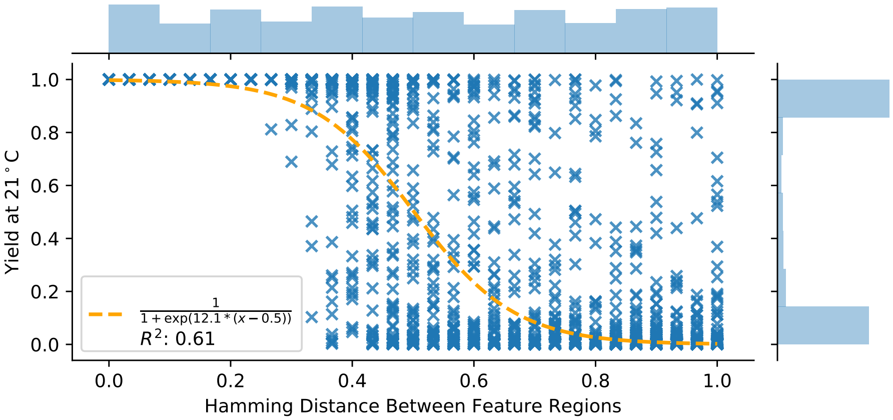
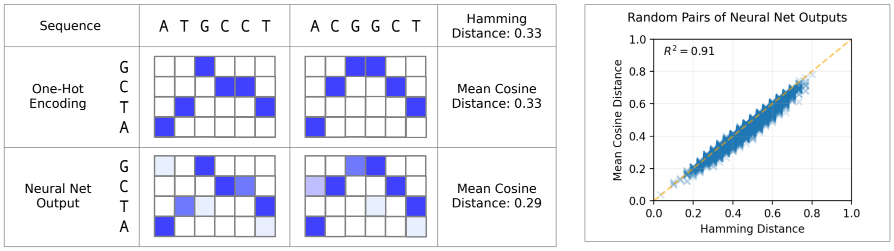


Fig. 4: Yield vs. Hamming distance for 2000 pairs of targets and queries with feature regions of length 30, as calculated by NUPACK. The dashed line shows the best sigmoid fit to the simulations.

However, Figure 4 shows that the query-target yield and the query-target Hamming distance have a noisy sigmoid[4] relationship. The best fit line provides us with a simple approximation of thermodynamic yield in terms of the Hamming distance. A drawback is that this approximation is less accurate for higher Hamming distances.

4.3 Approximating Hamming Distance

While we can use the Hamming distance to approximate thermodynamic yield, computing the Hamming distance requires discrete operations and is also not differentiable. Below, we define an alternative representation of DNA sequences, and a continuous approximation of Hamming distance that can be used with a neural network.



(a) Comparison of sequence representations and associated distance metrics.

(b) Effectiveness of neural network output.

Fig. 5: One-hot sequence encodings and their properties.

DNA sequences can be represented with a "one-hot" encoding, where each position is represented by a four-channel vector, and each channel corresponds to a base. For instance, if that base is an `A`, then the channel corresponding to `A` will have a value of one, and the other channels will be zero.

Figure 5a shows one-hot encodings of two sequences. At each position, the one-hot encodings can be compared by computing the cosine distance [5] between them. If they represent different bases, the representations will be orthogonal, and the cosine distance will be one. If they represent the same base, the cosine distance will be zero. Therefore the mean cosine distance across positions will be equal to the mean number of mismatches, which is equivalent to the Hamming distance.

A neural network cannot output differentiable representations that are exactly one-hot, because this would require discretization. However, if the channel values at each position are sufficiently far apart, we can approximate a one-hot

[4] Functions of the type:

$$f(x) = \frac{1}{1 + \exp(ax - b)}$$

[5] Given two vectors $\mathbf{u}$ and $\mathbf{v}$, the cosine distance is:

$$d(\mathbf{u}, \mathbf{v}) = 1 - \frac{\mathbf{u} \cdot \mathbf{v}}{||\mathbf{u}||\ ||\mathbf{v}||}$$

encoding by normalizing them with a softmax function[6], which pushes the maximum value towards one while pushing the other values towards zero. Furthermore, we can encourage the channel values to be far apart by using a hidden-layer activation function with a large output range, such as the rectified linear unit (ReLU) function[7].

Figure 5b shows the relationship between the mean cosine distance and Hamming distance of pairs of outputs, for 10,000 pairs of random inputs to a randomly initialized neural network with 10 input units, two ReLU hidden layers of 128 units each, and 30 four-channel softmax output units. The mean cosine distance between the neural network outputs closely follows the Hamming distance between their discretized counterparts, validating our approximation. To the best of our knowledge, using this four-channel encoding technique is a novel contribution of our work.

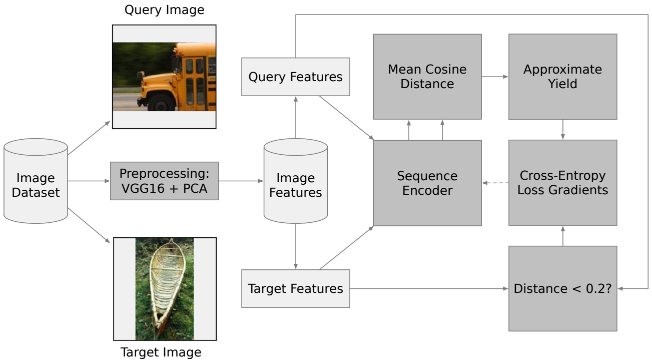


Fig. 6: The training loop, illustrating how a pair of images is used to calculate gradients for the sequence encoder. Data is in light gray, and operations are in dark gray.

4.4 Neural Network Architecture

Composing the yield approximation with the Hamming distance approximation allows us to use gradient descent to train any kind of neural-network-based sequence encoder to generate good encodings for similarity search, given a suitable dataset. This process is depicted in Figure 6. On each iteration, a pair of images is encoded, and then the mean cosine distance between the outputs is used

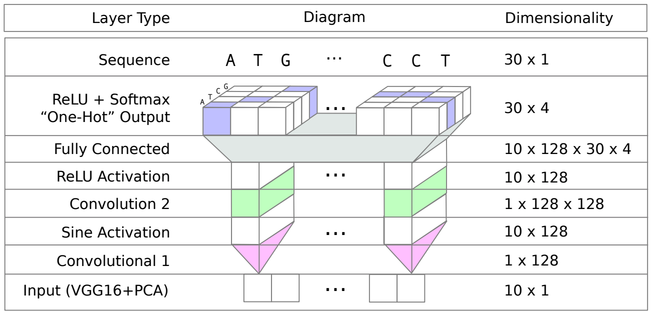
[6] Given an N-dimensional vector $\mathbf{u}$, the softmax function is defined element-wise as follows:

$$\text{softmax}(\mathbf{u})_i = \frac{e^{u_i}}{\sum_{j=1}^{N} e^{u_j}}$$

[7] The ReLU function is defined as:

$$\text{ReLU}(x) = \max(x, 0)$$

to calculate the approximate thermodynamic yield. Combined with the actual similarity between the feature vectors, the parameters of the neural network are updated using the gradient of the cross-entropy loss with respect to the parameters.



Layer Type	Diagram	Dimensionality
Sequence	A T G ··· C C T	30 x 1
ReLU + Softmax “One-Hot” Output	A T C G ···	30 x 4
Fully Connected		10 x 128 x 30 x 4
ReLU Activation	···	10 x 128
Convolution 2		1 x 128 x 128
Sine Activation	···	10 x 128
Convolutional 1		1 x 128
Input (VGG16+PCA)	···	10 x 1

Fig. 7: The neural network architecture for the sequence encoder.

A full exploration of the design space of neural-network-based sequence encoders is outside the scope of this work. We conducted a small-scale exploration and arrived at the architecture depicted in Figure 7, but this is not necessarily the best or only neural network for this task.

The network begins with two convolutional layers, where each input dimension is processed independently with a shared set of weights. This was done to preserve some of the “element-wise” structure of the Euclidean distance used to calculate the similarity label. The first convolutional layer has a sine-function activation, inspired by spectral hashing [26], a method for transforming an input feature space into a binary address space. The second convolutional layer uses the ReLU function to allow the outputs to be further apart.

Since the input dimensions do not have a spatial interpretation, we cap the convolutional layers with a set of fully connected weights to the four-channel sequence output, such that each input dimension’s activation map is given a chance to influence each base in all positions. A ReLU activation followed by a softmax activation gives us the approximate one-hot representation discussed above.

4.5 Training Results

To train the encoder, we first split the 30,607 images of the Caltech256 dataset into 24,485 training images and 6,122 test images. We extracted the VGG16 FC2 features from all 30,607 images, and then fitted a PCA transform to the FC2 vectors from the training set. The fitted transform was applied to all images.

During each training iteration, a batch of random pairs of training set images was used to update the encoder weights, as depicted in Figure 6. The encoder was trained for 65,000 iterations using 500 random pairs of images per iteration. Figure 8 shows the performance of the encoder as measured by the relationship between the thermodynamic yield (calculated with NUPACK) and the Euclidean

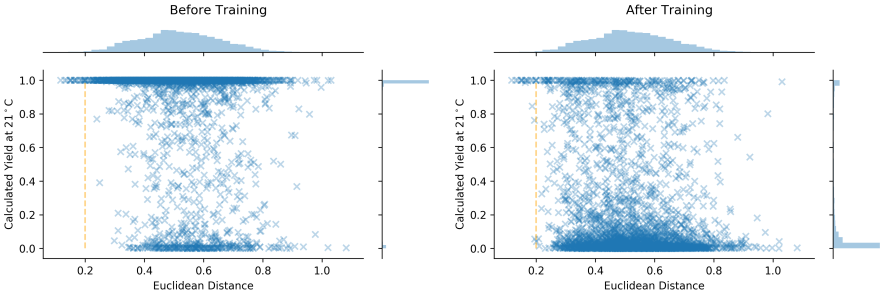


Fig. 8: Encoder performance on 3000 pairs of images from the test set, before and after training. The x-axis is the Euclidean distance between the target and the query, and the y-axis is the thermodynamic yield (calculated with NUPACK). The orange line shows the similarity threshold of 0.2.

distance between the images in the pair. NUPACK was set to simulate our experimental setup, with an equal molar ratio of target to query strands, and temperature at 21°C.

The performance is shown before training (with random parameters), and again after training. Before training, nearly all pairs of images exhibit a high yield, indicating no selectivity by distance. After training, most pairs have a low yield, but almost no pairs of images under 0.2 in Euclidean distance (which we have defined as similar) have a low yield. However, there are still non-similar images that have high yield, indicating that any successful query will also retrieve non-similar images.

5 Experiments

5.1 Dataset Construction

To test our designs in the wetlab, we constructed a subset of the test set consisting of 10 query images and 100 target images. The queries were chosen by first clustering all images in the training set into 10 groups using k-means, and then choosing a representative query image from the test set that belonged to each cluster. The k-means step ensures that none of the query images are pairwise-similar, because they all belong to different clusters in the data.

For each of these 10 query images, we selected its 10 nearest neighbors in the test set. This ensures that each query image has 10 similar images and 90 dissimilar images among the 100 targets. The result of this selection process is shown in Figure 9.

For each image, we encoded its features as a 30-nucleotide DNA sequence using the trained encoder, as described in Section 4.5. For each target image, we assigned it a random 5-nt ID, and then constructed a 90-nt sequence as shown in Figure 2a. For each query image, we constructed a 36-nt sequence as shown in Figure 2b. Target and query strands were then ordered from IDT. The query strands included the addition of a biotinylated spacer at the 5' end.

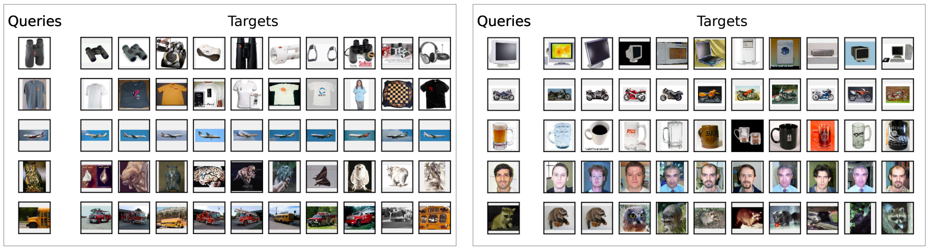


Fig. 9: The set of query and target images used in our wetlab experiments.

5.2 Target Preparation

All target strands were mixed together in an equal molar ratio. The targets were then mixed with 20% excess of the primer IP* at 10μM, and 20μL of the target-primer mixture was added to 20μL of 2x KAPA HIFI PCR enzyme mix. This 40μL mixture was placed in a thermocycler with the following protocol: (1) 95°C for 3 minutes, (2) 98°C for 20 seconds, (3) 56°C for 20 seconds, (4) 72°C for 20 seconds, (5) go to step 2 one more time, and (6) 72°C for 30 seconds. This process extends the primer to cover the 5' half of each target strand.

5.3 Query Protocol

For each of the 10 query strands, a sample of the target mixture was diluted to 200 nM and mixed with an equal molar concentration of the query, then annealed in a thermocycler from 95°C to 21°C at a rate of 1°C per minute.

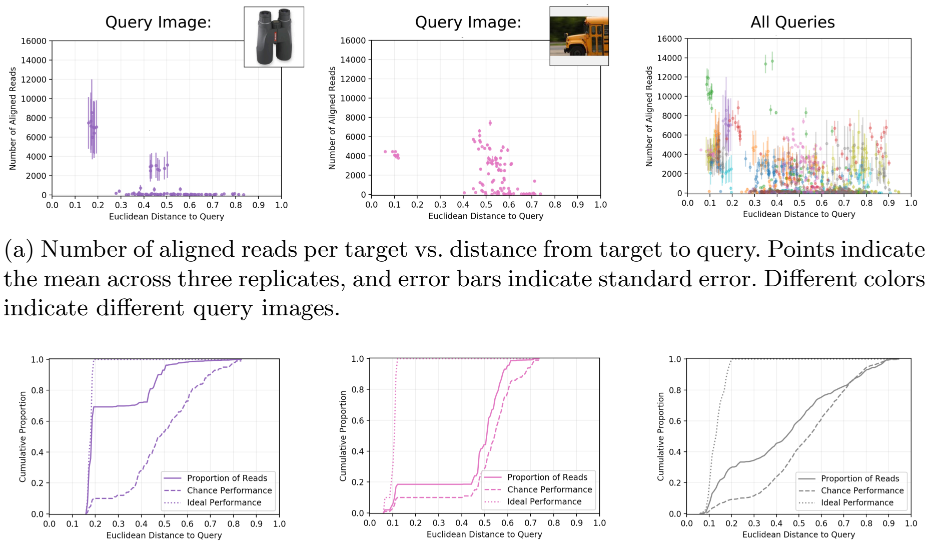
The annealed query-target mixture was mixed with streptavidin-coated magnetic beads, incubated at room temperature for 15 minutes, and placed on a magnetic rack. The supernatant containing non-captured DNA was removed and the beads were resuspended in elution buffer, then incubated for 5 min at 95°C and placed on a magnetic rack to separate captured DNA molecules from biotinylated query strands. The supernatant containing the captured DNA was mixed with the forward primer FP and the reverse primer RP* in a PCR reaction to amplify the captured targets. The amplified targets were ligated with Illumina sequencing adapters and then sequenced using an Illumina NextSeq.

This procedure was repeated 3 times for each of the 10 queries. Each query and replicate was given a unique sequencing index.

5.4 Results

For each query and replicate, the reads were aligned with the set of all target sequences using BWA-MEM [15]. Figure 10a shows the number of aligned reads for each target versus the distance from that target to the query, for two sample queries, and for all queries together.

Figure 10b shows the cumulative distribution of aligned reads as a function of distance from the query. The dashed line is a baseline indicating the cumulative



(a) Number of aligned reads per target vs. distance from target to query. Points indicate the mean across three replicates, and error bars indicate standard error. Different colors indicate different query images.

(b) Cumulative distribution of aligned reads as a function of increasing distance from target to query. For reference, the dashed lines show the cumulative distribution of targets by distance, and the dotted lines show an ideal where all reads are allocated to the nearest targets. Different colors indicate different query images (gray for all queries).

Fig. 10: Selected results for two of the ten query images, and aggregated results for all queries.

distribution of distances across the targets. The further the solid line is from the baseline, the stronger the relationship between distance and the number of reads. The dotted line shows the ideal result, where reads are only allocated to similar targets (those less than 0.2 Euclidean distance from the query).

The first sample query (the binoculars) shows a successful result, where most of the reads are allocated to similar targets. In contrast, the second sample query (the school bus) is less successful: the reads are distributed almost evenly across similar and non-similar images.

Across all queries, our results are moderately successful — though there are many reads going to dissimilar targets, our scheme is clearly capable of performing similarity-based enrichment: roughly 30% of the sequencing resources are being used by similar targets, which by construction make up just 10% of the database.

6 Discussion

In practice, the 10-dimensional image feature subspace used for our experiments is insufficiently selective. Referring back to Figure 1, the 100-dimensional space was more effective at relating distance to qualitative similarity. But it is diffi-

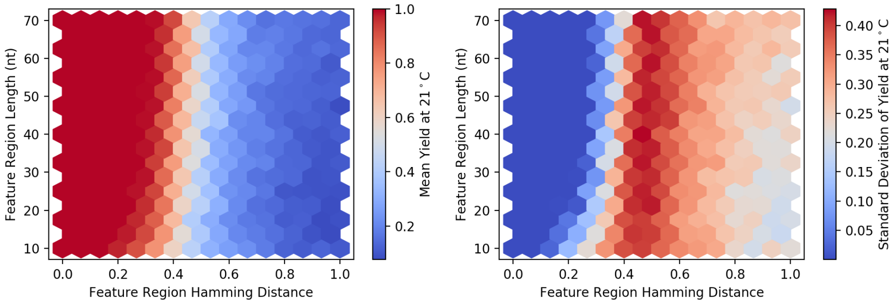


Fig. 11: Mean and standard deviation of yield as a function of feature region length and feature region Hamming distance.

cult to train an encoder to transform this already-compressed 100-dimensional subspace into a 30-nucleotide feature sequence.

We might be tempted to try longer feature regions, but this will likely experience more noise of the type seen in our results. Figure 11 illustrates this by generalizing Figure 4 to feature regions of different sizes. These plots bin across sequence length and target-query Hamming distance, and the color indicates either the mean (on the left) or the standard deviation (on the right) of the yield values in that bin, at our protocol temperature of 21°C. These plots tell us that selectivity decreases with increasing length, and that variance in yield for dissimilar targets increases as well.

These problems pose a difficult challenge to scaling this system. One avenue for future work is to devise a more accurate approximation for thermodynamic yield that can still be used to train a neural network. Another is to explore alternative probe designs that are meant to reduce variance, such as the toehold-exchange probes of Zhang et al. [27, 30].

7 Related Work

7.1 Content-addressable DNA databases

Baum was the first to propose DNA databases with associative search capabilities, noting the effectiveness of hybridization probes bound to magnetic beads [3]. Reif et al. designed and built a version of Baum's system. However, these schemes were meant to perform exact searching, while ours performs similarity search [20].

Reif and LaBean also proposed a scheme for performing similarity search in a Baum-style exact matching database [19]. This involved an *in silico* preprocessing step where the database was sorted into clusters. To retrieve similar data, a query would first be classified *in silico*, and everything in that cluster would then be retrieved *in vitro*. Because the cluster centers were static, a downside was that any data added to the database must be assigned to an existing cluster, which may not be accurate if the data belongs to a novel cluster that did not exist during training. In our system, the encoding also depends on a training set, but it is more flexible since there are no explicit cluster assignments.

7.2 Hybridization-Driven Similarity Search

Using melting temperature as a mechanism for similarity search in DNA databases was proposed by Tsaftaris et al. [22, 23]. However, their work focuses on similarity search of one-dimensional data, which allows the mapping from signal values to DNA sequences to be a small lookup table. Our system maps multidimensional input to DNA sequences.

Performing similarity search on higher dimensional data has been explored by Garzon and Neel [7, 16, 17]. Their work leverages a technique for *in vitro* dimensionality reduction of large datasets encoded in DNA (e.g., text corpora). On the other hand, our system performs dimensionality reduction *in silico* as part of the sequence encoding.

7.3 DNA Codeword Design

Designing codewords for robust DNA computing is a large subfield within the DNA computing community [13, 24]. These works focus on algorithms for computing a set of non-cross-hybridizing sequences that can be used to represent discrete signal values in an application-agnostic manner. Our system does not require non-cross-hybridizing sequences and takes an approach to codeword design, where the sequence mapping is learned from data.

8 Conclusion

We have presented a complete design, from encoding to sequencing, for a DNA database capable of performing content-based associative search by enriching database elements that are similar in content to a given query.

We have accomplished this by combining state-of-the-art research from the information retrieval and machine learning community with theoretical and experimental insights from the DNA computing and DNA storage communities to come up with novel encoding strategies and strand designs.

While it will be a challenge to scale this system to more complex features and larger datasets, this work is another step towards realizing the types of systems we will need to accommodate the storage demands of the future.

Acknowledgements We would like to thank the anonymous reviewers for their input, which were very helpful to improve the manuscript. We also thank the Molecular Information Systems Lab and Seelig Lab members for their input, especially Max Willsey, who helped frame an early version. We thank Dr. Anne Fischer for suggesting a better way to present some of the data. This work was supported in part by Microsoft, and a grant from DARPA under the Molecular Informatics Program.

Bibliography

[1] Adleman, L.M.: Molecular computation of solutions to combinatorial problems. Science **266**(5187), 1021–1024 (1994)

[2] Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. Communications of the ACM **51**(1), 117–122 (2008)

[3] Baum, E.B.: Building an associative memory vastly larger than the brain. Science **268**(5210), 583–585 (1995)

[4] Church, G.M., Gao, Y., Kosuri, S.: Next-generation digital information storage in DNA. Science **337**(6102), 1628–1628 (2012)

[5] Dirks, R.M., Bois, J.S., Schaeffer, J.M., Winfree, E., Pierce, N.A.: Thermodynamic analysis of interacting nucleic acid strands. SIAM Review (2007)

[6] Erlich, Y., Zielinski, D.: DNA Fountain enables a robust and efficient storage architecture. Science **355**(6328), 950–954 (2017)

[7] Garzon, M.H., Bobba, K.V., Neel, A.: Efficiency and reliability of semantic retrieval in DNA-based memories. DNA **2943**(Chapter 15), 157–169 (2003)

[8] Goldman, N., Bertone, P., Chen, S., Dessimoz, C., LeProust, E.M., Sipos, B., Birney, E.: Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. Nature **494**(7435), 77–80 (2013)

[9] Grass, R.N., Heckel, R., Puddu, M., Paunescu, D., Stark, W.J.: Robust chemical preservation of digital information on dna in silica with error-correcting codes. Angewandte Chemie Intl. Edition **54**(8), 2552–2555 (2015)

[10] Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Tech. rep., California Institute of Technology (2007)

[11] IDC: Where in the world is storage (2013), `http://www.idc.com/downloads/where_is_storage_infographic_243338.pdf`

[12] Indyk, P., Motwani, R.: Approximate nearest neighbors: Towards removing the curse of dimensionality. In: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing. pp. 604–613. STOC '98, ACM, New York, NY, USA (1998). https://doi.org/10.1145/276698.276876

[13] Kawashimo, S., Ono, H., Sadakane, K., Yamashita, M.: Dynamic neighborhood searches for thermodynamically designing DNA sequence. DNA Computing pp. 130–139 (2007)

[14] Lee, V.T., Kotalik, J., d. Mundo, C.C., Alaghi, A., Ceze, L., Oskin, M.: Similarity search on automata processors. In: 2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS). pp. 523–534 (2017)

[15] Li, H.: Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM (2013)

[16] Neel, A., Garzon, M.: Semantic Retrieval in DNA-Based Memories with Gibbs Energy Models. Biotechnology Progress **22**(1), 86–90 (2006)

[17] Neel, A., Garzon, M., Penumatsa, P.: Soundness and quality of semantic retrieval in DNA-based memories with abiotic data. In: 2004 Congress on Evolutionary Computation. pp. 1889–1895. IEEE (2004)

[18] Organick, L., Ang, S.D., Chen, Y.J., Lopez, R., Yekhanin, S., Makarychev, K., Racz, M.Z., Kamath, G., Gopalan, P., Nguyen, B., Takahashi, C.N., Newman, S., Parker, H.Y., Rashtchian, C., Stewart, K., Gupta, G., Carlson, R., Mulligan, J., Carmean, D., Seelig, G., Ceze, L., Strauss, K.: Random access in large-scale DNA data storage. Nature Biotechnology **36**(3), 242–248 (2018)

[19] Reif, J.H., LaBean, T.H.: Computationally inspired biotechnologies: Improved dna synthesis and associative search using error-correcting codes and vector-quantization? In: Condon, A., Rozenberg, G. (eds.) DNA Computing. pp. 145–172. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)

[20] Reif, J.H., LaBean, T.H., Pirrung, M., Rana, V.S., Guo, B., Kingsford, C., Wickham, G.S.: Experimental construction of very large scale dna databases with associative search capability. In: Jonoska, N., Seeman, N.C. (eds.) DNA Computing. pp. 231–247. Springer Berlin Heidelberg, Berlin, Heidelberg (2002)

[21] Salakhutdinov, R., Hinton, G.: Semantic hashing. International Journal of Approximate Reasoning **50**(7), 969–978 (2009)

[22] Tsaftaris, S.A., Hatzimanikatis, V., Katsaggelos, A.K.: DNA hybridization as a similarity criterion for querying digital signals stored in DNA databases. In: 2006 IEEE International Conference on Acoustics Speed and Signal Processing. pp. II–1084–II–1087. IEEE (2006)

[23] Tsaftaris, S.A., Katsaggelos, A.K., Pappas, T.N., Papoutsakis, T.E.: DNA-based matching of digital signals. In: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing. pp. V–581–4. IEEE (2004)

[24] Tulpan, D., Andronescu, M., Chang, S.B., Shortreed, M.R., Condon, A., Hoos, H.H., Smith, L.M.: Thermodynamically based DNA strand design. Nucleic Acids Research **33**(15), 4951–4964 (2005)

[25] Wan, J., Wang, D., Hoi, S.C.H., Wu, P., Zhu, J., Zhang, Y., Li, J.: Deep learning for content-based image retrieval: A comprehensive study pp. 157–166 (2014). https://doi.org/10.1145/2647868.2654948

[26] Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: Proceedings of the 21st International Conference on Neural Information Processing Systems. pp. 1753–1760. NIPS'08, Curran Associates Inc., USA (2008)

[27] Wu, L.R., Wang, J.S., Fang, J.Z., R Evans, E., Pinto, A., Pekker, I., Boykin, R., Ngouenet, C., Webster, P.J., Beechem, J., Zhang, D.Y.: Continuously tunable nucleic acid hybridization probes. Nature Methods **12**(12), 1191–1196 (2015)

[28] Yazdi, S.M.H.T., Gabrys, R., Milenkovic, O.: Portable and Error-Free DNA-Based Data Storage. Scientific Reports **7**(1), 1433 (2017)

[29] Zadeh, J.N., Steenberg, C.D., Bois, J.S., Wolfe, B.R., Pierce, M.B., Khan, A.R., Dirks, R.M., Pierce, N.A.: NUPACK: Analysis and design of nucleic acid systems. Journal of Computational Chemistry **32**(1), 170–173 (2011)

[30] Zhang, D.Y., Chen, S.X., Yin, P.: Optimizing the specificity of nucleic acid hybridization. Nature Chemistry **4**(3), 208–214 (2012)