Lal *et al. Genome Biology* (2025) 26:114
https://doi.org/10.1186/s13059-025-03584-9

Genome Biology

SOFTWARE Open Access

Polygraph: a software framework for the systematic assessment of synthetic regulatory DNA elements

Avantika Lal[1†], Laura Gunsalus[1†], Anay Gupta[2], Tommaso Biancalani[1] and Gokcen Eraslan[1*]

[†]Avantika Lal and Laura Gunsalus contributed equally to this work.

*Correspondence: eraslan.gokcen@gene.com

[1] Biology Research | AI Development, gRED Computational Sciences, Genentech, South San Francisco, USA
[2] College of Computing, Georgia Institute of Technology, Atlanta, GA, USA

Abstract

The design of regulatory elements is pivotal in gene and cell therapy, where DNA sequences are engineered to drive elevated and cell-type specific expression. However, the systematic assessment of synthetic DNA sequences without robust metrics and easy-to-use software remains challenging. Here, we introduce Polygraph, a Python framework that evaluates synthetic DNA elements, based on features like diversity, motif and *k*-mer composition, similarity to endogenous sequences, and screening with predictive and foundational models. Polygraph is the first instrument for assessing synthetic regulatory sequences, enabling faster progress in therapeutic interventions and improving our understanding of gene regulatory mechanisms.

Keywords: Sequence design, Synthetic biology, Machine learning, Sequence modeling, Regulatory genomics

Background

Cis-regulatory DNA elements (CREs) are DNA sequences that drive and tune gene expression, in part through the selective binding of transcription factors or other regulatory molecules necessary to initiate and maintain transcription [1]. Their cell-type specific activity makes carefully designed CREs a promising avenue for synthetic biology and DNA-based nucleic acid therapeutics such as cell and gene therapy [2–6]. Recent deep learning models trained on high-throughput sequencing data integrate the complex grammar embedded in DNA and serve as powerful tools to predict the activity of native and synthetically generated regulatory elements [7–11]. However, evaluating and prioritizing designed sequences remains challenging without a comprehensive understanding of their properties and potential regulatory mechanisms.

Current DNA design methods range from classic optimization algorithms [12–15] to generative modeling approaches [16–26] aimed at driving desired expression profiles, transcription factor binding patterns, and other functional outcomes. These methods invite exciting open questions: What regulatory mechanisms do they exploit? Do they

converge on the regulatory grammar observed in natural sequences or identify novel combinations of binding motifs? How should synthetic sequences be selected from millions of computationally generated options? Our ability to answer these questions is currently limited by the lack of integrated software to predict and analyze CRE performance across biological contexts, and by the absence of defined metrics for CRE evaluation, making it difficult to standardize assessments across studies and design methods.

Here, we present Polygraph, a software package that enables systematic evaluation and selection of designed DNA sequences through sequence analysis, transcription factor motif composition analysis, embedding analysis, predictive modeling, and language modeling. By facilitating the comparison of design algorithms and prioritization of CRE candidates, Polygraph provides a path toward robust and interpretable CRE engineering.

Results

The Polygraph package

Polygraph is a Python package that accepts DNA sequences of any length (Fig. 1A), analyzes their properties, and compares them to user-defined reference sequences such as genomic regulatory elements. Polygraph also supports statistical significance testing

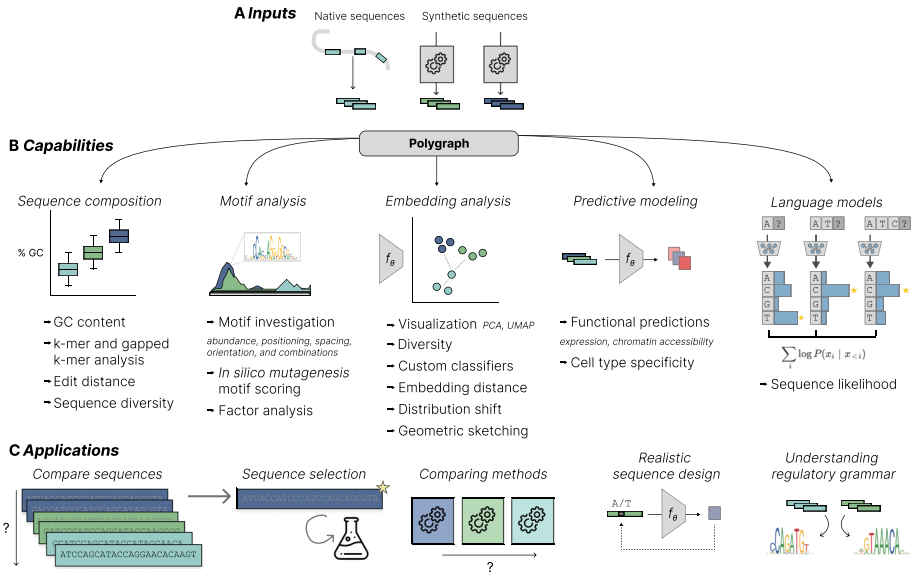

Fig. 1 Schematic of Polygraph. Polygraph is a comprehensive package to evaluate and compare DNA sequences. **A** Inputs: Polygraph accepts both native DNA and synthetic sequences for analysis. **B** Capabilities: Five classes of analysis including sequence composition (GC content, *k*-mer and gapped *k*-mer abundance, and edit distance between synthetic sequences and reference sequences); motif analysis (quantifying transcription factor binding sites, analyzing motif combinations, positioning, and orientation, scoring motifs with in silico mutagenesis (ISM), and performing non-negative matrix factorization (NMF) to identify common transcription factor programs); embedding analysis, where sequences are embedded in a low-dimensional space based on *k*-mer/motif content or deep learning model latent spaces, which are used to visualize sequences, compute group diversity, and calculate between-group distances; predictive modeling to evaluate designed sequences based on activity, specificity, and chromatin accessibility; and application of a human DNA language model to score sequence likelihood. **C** Applications: Polygraph enables users to compare sequences generated by different methods, select diverse and/or realistic synthetic sequences, design realistic sequences with guided evolution, and study the regulatory grammar of synthetic sequences

across groups and predictions using custom models (Fig. 1B). Below, we summarize the features available in the package.

Sequence composition analysis

Polygraph includes tools to evaluate sequence composition metrics like GC content, *k*-mer frequencies (abundance of nucleotide subsequences of length *k*), and edit distance between synthetic sequences and reference sequences. Human regulatory regions exhibit elevated GC content [27], enrichment of functional subsequences like GATA [28], CpG islands [28], and short repeats [29]. Evaluating sequence composition with our quantitative metrics could therefore provide insight into the novelty and "humanness" of computationally designed regulatory elements.

Transcription factor binding motif analysis

Motif content reveals which combinations of transcription factors can bind to a DNA sequence. Polygraph scans sequences with TF binding motifs, reports the number of motif matches, and computes the enrichment of motifs and pairs of motifs in synthetic sequences compared to reference sequences. It also analyzes the start positions, orientations, and spacing of motifs, indicating if placed motifs are centered and if different methods pursue different syntactic strategies. Further, Polygraph applies non-negative matrix factorization (NMF) to decompose the motif count matrix into common transcription factor programs shared across sequences [15]. Model-driven in silico mutagenesis can be used to score the importance of each motif. Altogether, motif analysis may uncover higher-order regulatory rules exploited by different design approaches.

Embedding analysis

Polygraph enables users to create an embedding of a set of DNA sequences, in which each sequence is represented either by its content of *k*-mers, gapped *k*-mers, or motifs, or by its latent embedding created by the lower layers of a neural network model. Such model-based representations may capture higher-order interactions which are not obtained by *k*-mer or motif representations alone. Regardless of which sequence embedding is chosen, the user can then perform the following analyses:

(1) Visualize sequences in the embedding space with PCA or UMAP [30].
(2) Compute a *sequence diversity* metric, defined as the average *k*-nearest neighbor (KNN) distance between a sequence and its neighbors from the same group [15], to quantify how similar designed sequences are to each other.
(3) Train support vector machine (SVM) classifiers to empirically test whether synthetic sequences can be discriminated from native DNA.
(4) Compute Euclidean and *k*-nearest neighbor distances between sequence groups.
(5) Perform a statistical test of distribution shift between sequence groups in the embedding space.
(6) Geometric sketching [31] to sample a representative subset of sequences based on their spacing in the embedding space.

Predictive modeling

Polygraph integrates trained neural network models to evaluate designed sequences on key properties like activity, specificity, and chromatin accessibility. We provide three pretrained open-source models for yeast and human prediction, with the option of integrating custom PyTorch models. Polygraph enables flexible and fast model prediction on generated sequences, as well as cell-type specificity evaluation.

Language modeling

Autoregressive DNA language models are self-supervised models that aim to implicitly capture the rules of gene regulation as well as the rules of coding genes of the genome on which they are trained. These models can be queried to infer how likely a given sequence is to be sampled from the training set, which, in the case of human language models, is the human genome. Here, we use HyenaDNA [32] to quantify the log-likelihood of synthetic sequences. These scores represent their "humanness" which is used as a proxy metric of how realistic generated sequences are.

Guided evolution

We provide a "guided evolution" function to evolve DNA sequences with high predicted activity while maintaining similarity to reference native sequences, based on their Euclidean distance in an embedding space (see Methods).

Polygraph enables multiple downstream applications, including comparing sequences generated by different methods, selecting diverse and/or realistic synthetic sequences, designing realistic sequences with guided evolution, and studying the regulatory grammar of synthetic sequences (Fig. 1 C). In the next two sections, we demonstrate the use of this package on two datasets: yeast promoters designed with directed evolution [33], first-order optimization method Ledidi [13], and guided evolution, as well as human enhancers designed by Gosai et al. [15] using FastSeqProp [12], simulated annealing [34], and AdaLead [14].

Polygraph identifies divergent regulatory strategies in native and computationally designed yeast promoters

One of the most common strategies for regulatory element design [12, 15, 35] is to start with natural or random DNA sequences with sub-optimal function and use a sequence-to-function model to iteratively edit these until the model predicts that the desired function is achieved. However, these methods do not constrain the composition of the regulatory sequences to be similar to natural DNA, raising the question of whether they yield different regulatory strategies from those found in nature.

We selected 50 native yeast promoters with low measured activity in media ("Native (Weak)") [36, 37], as starting points for design. Using a convolutional regression model trained to predict the promoter activity of native yeast DNA, we iteratively edited each sequence to increase its activity. We used three methods: directed evolution [33] ("Directed Evolution"), the gradient-based optimization method Ledidi [13] ("Gradient"), and Polygraph's guided evolution function which maximizes predicted activity while maintaining similarity to native sequences ("Guided Evolution"; see

Methods). Each method produced a set of 50 putative strong promoters, one based on each weak promoter in the starting set. Finally, we collected the 50 native yeast promoters with the highest measured activity in media ("Native (Strong)") as a reference set. All three design methods produced synthetic promoters with similar levels of predicted activity, comparable to or higher than the native strong promoters (Additional file 1: Fig. S1).

We first analyzed promoters based on their sequence composition. Native (Weak) promoters had a mean edit distance of 40.2 from Native (Strong) promoters. This was not significantly altered after editing by Directed Evolution or Gradient methods (Directed Evolution mean = 41.0, Gradient mean = 40.4, Kruskal–Wallis test p value 4×10^{-5}, Dunn's post hoc test p value, Directed Evolution vs. Native (Weak) = 0.5, Gradient vs. Native (Weak) = 0.1), suggesting that these sequences may not grow to more closely resemble effective native promoters through iterative design (Fig. 2A). On the other hand, Guided Evolution produces sequences more similar to Native (Strong) (mean = 39.0, Dunn's post hoc test p value = 0.007).

Further examination shows a dramatic difference between sequences generated by Guided Evolution versus other approaches. While both strong and weak native promoters had a mean GC content of approximately 0.39, consistent with the composition of the *Saccharomyces cerevisiae* genome [38], Directed Evolution and Gradient methods produced significantly more GC-rich sequences (Directed Evolution mean = 0.45, Gradient mean = 0.49, Kruskal–Wallis test p value $= 5.2 \times 10^{-12}$, Dunn's post hoc test p value $= 1.3 \times 10^{-4}$ for Directed Evolution vs. Native (Strong), 10^{-7} for Gradient vs. Native (Strong)). In contrast, sequences produced by Guided Evolution had GC content similar to native DNA (mean = 0.39, Dunn's post hoc test p value = 0.93) (Fig. 2B).

Next, we calculated the frequencies of all 4-mers in all promoter sequences and compared them between groups using a two-sided Mann–Whitney U test with Native (strong) promoters as the reference group. We defined 4-mers with FDR-adjusted p value < 0.01 as differentially abundant. Promoters designed by Directed Evolution and Gradient were significantly enriched for numerous 4-mers (Fig. 2C, Additional file 1: Table S1) including GC-rich patterns like "CGCG." They were also enriched in some AT-rich patterns such as TGAT and GAAT, which were also enriched in Native (Weak) promoters. This demonstrates that these methods both add unrealistic sequence patterns and retain some patterns from their initial sequences that are uncharacteristic of strong native promoters. In contrast, only 5 4-mers were significantly enriched or depleted in the Guided Evolution promoters (Additional file 1: Table S1).

We scanned all sequences with yeast transcription factor (TF) binding motifs from JASPAR [39]. Once again, Directed Evolution and Gradient promoters differed significantly in motif content from Native (Strong) promoters, whereas Guided Evolution did not (Fig. 2D, Additional file 1: Table S2). Specifically, Directed Evolution promoters were enriched for SWI4, RSC3, MBP1, and MBP1-SWI6 motifs, and Gradient promoters were enriched for these as well as RSC30, SUT1, DAL81, CHA4, PUT3, and TEA1 motifs. These motifs were rarely seen in Native (Weak) promoters, indicating that they were introduced in the design process. In fact, SWI4 and CHA4 motifs were absent from all native strong or weak promoters, arising exclusively in the optimization process. Most of these TFs (SWI4, MBP1, MBP1-SWI6, DAL81, CHA4, PUT3, and TEA1) are known

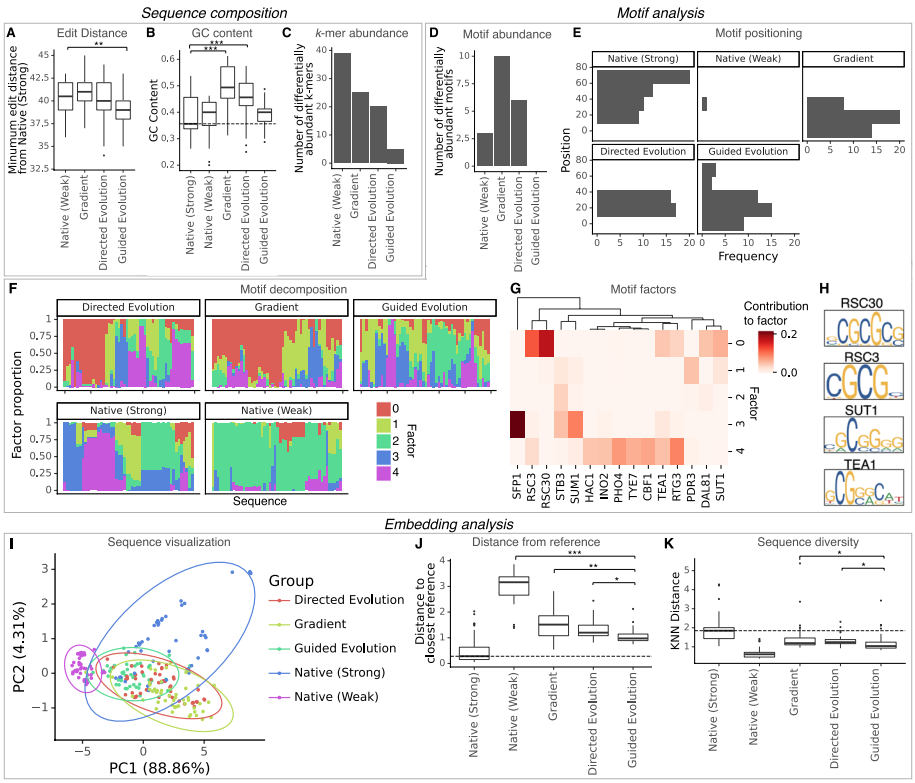

Fig. 2 Sequence analysis of native and designed yeast promoters. **A** Box plots showing the distribution of the edit distance from the most similar strong native yeast promoter, for weak native yeast promoters (Native (Weak)), and synthetic promoter sequences designed by editing Native (Weak) promoters via directed evolution (Directed Evolution), gradient-based optimization (Gradient), or guided evolution (Guided Evolution). **B** Box plots showing the distribution of GC content in each group of sequences. Native (Strong) represents strong native yeast promoters. **C** Bar plot showing the number of differentially abundant 4-mers (Mann–Whitney *U* test FDR-adjusted *p* value < 0.01) in each group of promoters compared to Native (Strong). **D** Number of differentially abundant transcription factor binding motifs (Mann–Whitney *U* test FDR-adjusted *p* value < 0.01) in each group of promoters compared to Native (Strong). **E** Histogram of SFP1 motif start locations in each group of promoters. **F** Non-negative matrix factorization (NMF) of the motif frequency matrix into 5 components. Each column represents a sequence and colors show the contribution of each factor to the motif composition of the sequence. **G** Heatmap showing the top motifs and their contributions to each NMF component. The top 15 motifs ranked by maximum contribution to any NMF component are shown. **H** Top 4 motifs enriched in factor 0. **I** PCA visualization of sequence embeddings from the last convolutional layer of a sequence-to-expression predictive model, for all groups of promoter sequences. Ellipses represent the 95% confidence boundary of multivariate normal distributions fitted to the data. **J** Box plots showing the distance of each sequence to its nearest neighbor in the Native (Strong) group, in the embedding space shown in **I**. **K** Box plots showing the Euclidean distance of each sequence to its 5 nearest neighbors in the same group in the embedding space shown in **I**, a metric of within-group sequence diversity. *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$

transcriptional activators [40–45], while RSC3 and RSC30 can indirectly increase promoter activity through chromatin remodeling [46]. Classifiers trained to distinguish synthetic from native promoters based on motif content were least successful with Guided Evolution promoters, further highlighting their comparatively realistic composition (Additional file 1: Fig. S2).

While many studies evaluate designed sequences on the presence of relevant TF motifs, Polygraph enables more complex analyses of regulatory grammar. For example, the SFP1 motif is found in native strong promoters and in all groups of synthetic

promoters and is not significantly different in abundance. However, all design methods tended to introduce this motif between positions 0–40 of the sequence, in contrast to native strong promoters which tend to contain this motif between positions 40–80 (Fig. 2E). In fact, each group of designed promoters had 6–10 motifs with significant differences in positioning relative to native promoters (Additional file 1: Table S3). Specific motif pairs like DAL81/RSC30, RSC3/MBP1, and SUT1/MBP1 were also significantly more likely to co-occur in designed groups than in native strong promoters (Additional file 1: Table S4).

We used Polygraph's NMF function to decompose the motif frequency matrix into 5 factors, or patterns of co-occurring motifs that best explain the variation in the data. Strikingly, factor 0 is rare in both native and Guided Evolution promoters, but is created in Directed Evolution and Gradient promoters (Fig. 2F, Additional file 1: Fig. S3, Additional file 1: Table S5; Mann–Whitney U test FDR-adjusted p value Directed Evolution vs. Native (Strong) $= 1.2 \times 10^{-3}$, Gradient vs. Native (Strong) $= 3.4 \times 10^{-12}$, Guided Evolution vs. Native (Strong) $= 0.08$). This factor is dominated by several of the activating motifs mentioned above, including RSC30, RSC3, SUT1, and TEA1 (Fig. 2G)—all of which are GC-rich (Fig. 2H).

Finally, we embedded all the promoter sequences in a low-dimensional space using an independent model trained to predict yeast expression from sequence (Fig. 2I, Additional file 1: Fig. S4). Model-based embeddings implicitly capture many relevant features, including motif composition, spacing, positioning, and co-occurrence. In this embedding space, Guided Evolution promoters are the closest to native strong promoters (Fig. 2J; mean distance to the closest Native (Strong) sequence, Native (Weak) $= 2.99$, Directed Evolution $= 1.33$, Gradient $= 1.51$, Guided Evolution $= 1.04$; Kruskal–Wallis test p value $= 6.4 \times 10^{-34}$; Dunn's post hoc test p values, Directed Evolution vs. Guided Evolution $= 6.4 \times 10^{-3}$, Gradient vs. Guided Evolution $= 3.2 \times 10^{-4}$). This is true irrespective of the model layer used to embed the sequences (Additional file 1: Fig. S5). Also, all groups of synthetic promoters were significantly less diverse than native strong promoters as measured by the inter-group KNN distance in embedding space, with Guided Evolution being the least diverse (Fig. 2K; Kruskal–Wallis test p value $= 2.5 \times 10^{-27}$, Dunn's post hoc test p value: Gradient vs. Native (Strong) $= 3.4 \times 10^{-4}$; Directed Evolution vs. Native (Strong) $= 6.1 \times 10^{-4}$; Guided Evolution vs. Native (Strong) $= 9.3 \times 10^{-9}$).

We repeated these analyses on promoters designed by editing random DNA sequences instead of weak native promoters and found similar results (Additional file 1: Fig. S6). In sum, regardless of the initial sequences used, Directed Evolution and Gradient methods adopted a regulatory grammar based on GC-rich activating motifs, which is rare in native sequences, potentially due to other genomic constraints. For example, high GC content drives higher rates of mutation and recombination in yeast genomes [47]. In contrast, guided evolution produced sequences more similar to native strong promoters, while still achieving high predicted activity. Therefore, incorporating Polygraph metrics into existing design methods can produce effective sequences that also mimic natural regulatory grammar. However, this method still has limitations, notably low diversity of the generated sequences, and differences in positioning and co-occurrence of some motifs. This example illustrates how Polygraph's in silico metrics can be used to evaluate and compare design methods as well as improve the design process.

Polygraph identifies functionally diverse subsets of computationally designed HepG2 enhancers

We next applied Polygraph to evaluate ~16,000 synthetic human enhancer sequences designed for the HepG2 cell line using three different methods: AdaLead, FastSeqProp, and simulated annealing. These generative approaches were guided by an expression prediction model trained on a massively parallel reporter assay (MPRA) of 776,474 genomic sequences in three cell lines. In addition, the authors analyzed ~8000 native human regulatory elements with HepG2-specific regulatory activity [15].

We first compared the computationally designed enhancers to the native human sequences. While all groups of sequences had a mean GC content close to 50%, designed sequences had lower GC variance (Additional file 1: Table S6). All groups of synthetic sequences were clearly distinguished from native sequences based on *k*-mer and motif content (Additional file 1: Table S7). In addition, we computed sequence likelihoods using an autoregressive DNA language model [32]. All groups of synthetic sequences showed significantly lower log-likelihood than native sequences which were in the test set of the language model (Fig. 3A; Mann–Whitney *U* test FDR-adjusted *p* value $< 10^{-250}$ for all methods), showing once again their difference from the human genome.

We used NMF to identify 5 regulatory programs representing co-occurring combinations of TF binding motifs. For simplicity, we restricted the motif count matrix to motifs for TFs that are expressed in the human liver as per GTEx (Fig. 3B–C). The top motifs contributing to each factor are TFAP4, ASCL1, and TCF4 in factor 0; AP-1 (FOS/JUN) and CEBPG motifs in factor 1; TFAP2 A/2 C motifs in factor 2; HNF4 A, RXRG, and NR2 F1 motifs in factor 3; and HNF1 A/B motifs in factor 4 (Fig. 3C). Of these, HNF4 A, HNF1 A, HNF1B, and proteins of the C/EBP family are known activators of gene expression in hepatocytes [48], and AP-1 is also involved in hepatocyte proliferation [49]. All groups of synthetic sequences were significantly enriched for factors 0 and 3 in comparison to native sequences, and depleted for factors 1 and 2, demonstrating the differential use of liver-specific regulatory syntax (Fig. 3B, Additional file 1: Fig. S7, Additional file 1: Table S8).

Polygraph includes a binary sequence-to-chromatin accessibility model trained on 203 cell types from 30 adult and 15 fetal human tissues (Additional file 1: Fig. S8). Based on this model, the cell types with the highest predicted accessibility for the native elements are fetal hepatoblasts and adult hepatocytes, which are most similar to the HepG2 cell line. All groups of designed sequences were predicted to be highly accessible in these cell types, even more so than native sequences (mean prediction in fetal hepatoblasts: AdaLead = 0.67, simulated annealing = 0.77, FastSeqProp = 0.79, native = 0.42, Fig. 3D; Mann–Whitney *U* test *p* value, native vs. synthetic predictions in fetal hepatoblasts $< 10^{-250}$). However, synthetic sequences also had similarly high predicted accessibility in other cell types such as enterocytes, colon epithelial cells, and pancreatic acinar cells, which were not considered in the design process (Fig. 3D, Additional file 1: Fig. S9). Both native and synthetic sequences had low predicted accessibility in unrelated cell types such as immune and neuronal cell types. This shows how Polygraph provides orthogonal validation of synthetic sequences and can identify potential side effects in diverse cell types. We also embedded the sequences using the intermediate layers of this model (Fig. 3E). Consistent with the *k*-mer and motif embeddings, the model embedding shows

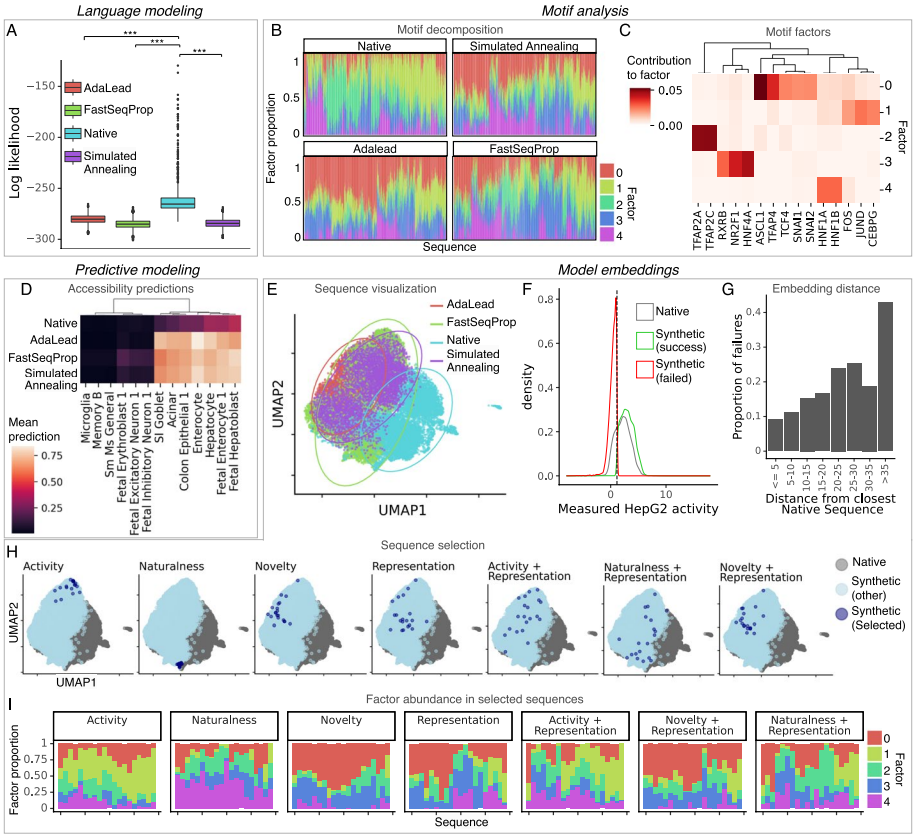

Fig. 3 Sequence analysis of designed and native human enhancers in HepG2 cells. **A** Box plots showing the distribution of log-likelihoods of native and synthetic HepG2 enhancers designed by three different methods. Log-likelihood was calculated using an autoregressive language model trained on human genomic DNA. Native sequences were restricted to those in the test set of the language model. ***: $p < 0.001$. **B** NMF of the motif frequency matrix. Each column represents a sequence and colors show the contribution of each factor to the motif composition of the sequence. **C** Heatmap showing the top motifs contributing to each NMF component. **D** Heatmap showing the average predicted probability of accessibility in selected cell types, for native enhancers as well as synthetic enhancers designed by three different methods. Predictions were made with a binary classification model trained to predict chromatin accessibility in 203 human cell types. A subset of cell types with the highest and lowest predicted accessibility is shown. **E** UMAP visualization of sequence embeddings from the sequence-to-accessibility predictive model used in **A**. Ellipses represent the 95% confidence boundary of multivariate normal distributions fitted to the data. **F** Density plot showing the experimentally measured activity in HepG2 cells, for native enhancers (gray) and synthetic enhancers. Synthetic enhancers are separated into experimental successes (green) and failures (red). The dashed line represents the threshold to separate successes and failures. **G** Bar plots showing the fraction of sequences that failed in experimental validation, grouped by their Euclidean distance to the most similar native enhancer in the embedding space shown in **E**. **H** UMAP visualization of sequence embeddings as shown in **E**. Dark blue dots represent sets of 20 synthetic enhancers, selected using different criteria. **I** Partitioning of the motif frequencies of the sets of 20 synthetic enhancers shown in **G** into the 5 regulatory programs shown in **B**, **C**

a clear difference in the distribution of synthetic and native sequences (Hotelling's T2 test p value $< 10^{-250}$). An SVM classifier based on embedding values could distinguish between native and synthetic sequences with an AUROC of 0.99.

A major reason for concern about the realism of synthetic elements is that extremely unnatural sequences may be incorrectly predicted due to being out-of-distribution for predictive models, and hence may fail in experimental validation. Since Gosai et al.

experimentally validated their synthetic enhancers [15], we tested this hypothesis by dividing the synthetic enhancers into successes and failures based on their experimentally measured activity in HepG2 cells (Fig. 3F). While most synthetic designs were successful, the proportion of failures increased with the distance from native sequences in embedding space (Fig. 3G), indicating that sequences with very different composition from natural regulatory elements are more likely to fail. This highlights the importance of Polygraph's metrics of sequence realism.

A related, common problem is how to select a limited number of synthetic sequences for experimental validation. For instance, we might need to select only 20 out of the ~16,000 synthetic enhancers in this study for validation. An obvious approach would be to select the 20 sequences with the highest predicted activity. However, this subset ("Activity") converges on a single regulatory strategy dominated by factor 1 (Fig. 3H–I). Polygraph's metrics can help select sets of sequences based on different criteria depending on the user's goals. For example, we selected 20 sequences based on "Naturalness" (sequences with the least distance from any native enhancer). This approach is safe, choosing a set of sequences that resulted in no experimental failures (Additional file 1: Fig. S10) and followed a regulatory strategy dominated by factor 4, similar to a subset of native elements (Fig. 3I). On the other hand, users may be interested specifically in validating novel regulatory syntax, so we also selected 20 sequences based on "Novelty" (i.e., sequences furthest from native elements in embedding space). These are dominated by factors 0 and 3, which are significantly less abundant in native sequences (Fig. 3 B, I); however, this strategy risks experimental failure (Additional file 1: Fig. S10).

Polygraph implements geometric sketching [31] to select a set of sequences that compactly represents the regulatory landscape. We used this to select 20 representative sequences ("Representation"; Fig. 3H) which spanned diverse regulatory strategies (Fig. 3I). This can also be combined with other measures; for example, we used geometric sketching to select 20 sequences among the 5% sequences with highest activity ("Activity + Representation"), the 5% sequences with highest Naturalness ("Naturalness + Representation"), and the 5% sequences with highest Novelty ("Novelty + Representation"). In each case, we observed that geometric sketching diversifies the selected regulatory strategies while preserving the property of interest (Fig. 3H–I, Additional file 1: Fig. S11).

Discussion

The rational design of regulatory DNA elements holds great promise for both improving our understanding of gene regulatory mechanisms and driving cell-type specific expression in therapeutic contexts. Recent studies have provided a wealth of new methods and designed elements with cell type- and tissue-specific activity. However, in most cases the evaluation of these sequences is limited to examining the presence of known regulatory elements. Few studies have used systematic quantitative metrics to evaluate synthetic sequences, and those that do have used widely different metrics making systematic comparison difficult (Additional file 1: Table S9).

Polygraph provides an integrated toolkit for evaluating designed regulatory elements. As demonstrated in yeast and human case studies, Polygraph provides insights into the

properties of computationally designed regulatory elements. Sequence analysis revealed how different generative algorithms converge on common grammar, often distinct from native DNA. Motif scanning uncovered shared transcription factor binding sites exploited for specialized activity like cell-type specific expression across methods. Predictive neural networks allow in silico validation of synthetic elements in diverse biological contexts and provide an informative latent representation of sequences based on high-level regulatory sequence features. DNA language modeling provides a "humanness" score describing similarity to genomic sequences. Incorporating these similarity metrics into "guided evolution" yielded synthetic regulatory elements predicted to have high activity while reducing extreme divergence from natural sequences. These metrics together with geometric sketching enable selection of diverse synthetic sequences with desired properties for experimental validation.

Together, Polygraph enables comprehensive assessment and comparison of regulatory elements and design approaches. This was recently demonstrated by the use of Polygraph to evaluate synthetic cis-regulatory elements designed using an autoregressive language modeling approach, demonstrating that the language model-generated elements were more realistic than those produced by predictive model-guided approaches based on a wide variety of metrics [23].

Several exciting directions remain to improve analysis and application. Integrating a wider array of predictive models, including those predicting protein binding, histone modifications, and chromatin states, will add more regulatory perspectives to sequence evaluation. Experimentally validating a small number of synthetic sequences and iteratively feeding the results back to the generative design processes in an active learning framework is a promising future direction.

Methods

Polygraph package details

Sequence composition analysis: Polygraph calculates the following metrics of sequence composition for each provided sequence: sequence length, GC content, edit distance with respect to reference sequences, *k*-mer frequency, gapped *k*-mer frequency, unique *k*-mers, and *k*-mer positions. Embedding metrics (described below) can be computed for each group of sequences based on their *k*-mer frequency. It uses statistical tests (described below) to compare the distribution of any of these metrics across groups.

Motif analysis: Polygraph uses FIMO [50] to perform motif scanning on all provided sequences with either the JASPAR databases [39] or a set of user-provided motifs. Based on the FIMO results, it computes the frequency of each motif, frequency of pairs of motifs, and the relative orientation and spacing between pairs of motifs. Embedding metrics (described below) can be computed for each group of sequences based on their motif frequency. Statistical tests (described below) are used to compare the distributions of these metrics across groups, resulting in statistics for differential motif abundance, differential positioning, differential orientation, and differential spacing. Polygraph uses NMF to perform factor analysis on the motif frequency matrix.

One issue with motif analysis using the full JASPAR database is that this may include motifs for TFs that are not relevant to the tissue or cell type being analyzed. Polygraph allows users to filter motifs based on tissue-specific expression of the corresponding TF in GTEx [51]. Also, given a predictive model, Polygraph can perform in silico mutagenesis (ISM) scoring for each base in each input sequence, and rank motifs by their average importance score (change in prediction upon mutating a base in the motif) across all sequences, indicating their potential relevance to the system under study. The user can then select the top motifs based on this ranking and restrict subsequent analyses to this set. This can allow the user to identify the motifs most relevant to biological activity in the experimental context, which may enable more meaningful and interpretable results in downstream analyses. However, this will necessarily focus only on motifs relevant to whatever targets were included in the training set of the provided model. Users may use the trained models provided with the package, or any custom PyTorch model, for this purpose.

Predictive modeling analysis: Polygraph facilitates predictive modeling analysis through trained deep learning models, such as Enformer [10], our pre-trained yeast and human models, or custom user-provided PyTorch models. The package allows for generating model predictions and calculating cell-type specificity. Cell-type specificity is estimated by the MinGap metric [15], i.e., the difference between the minimum predicted activity in on-target cell types and the maximum predicted activity in off-target cell types.

Language models: Polygraph supports computing sequence likelihood via the HyenaDNA family of autoregressive models [32].

Embedding analysis: Polygraph embeds sequences based on their *k*-mer content, motif content, or using the intermediate layers of a deep learning model. Enformer [10], Nucleotide Transformer [52], our pre-trained yeast and human models, or any custom user-provided PyTorch models can be used for this purpose, and the user can choose the model layer from which to output embeddings. The following embedding metrics can be computed for each group of sequences based on any of these embeddings.

1. Sequence diversity: Calculated as the within-group KNN distance, representing the mean Euclidean distance of each sequence to its *k*-nearest neighbors within the same group.
2. Distance to reference: Measures the Euclidean distance of each sequence to its nearest neighbor in the reference group, within the embedding space.
3. 1-NN statistic: Computes the fraction of synthetic sequences in each group that have a reference sequence as their nearest neighbor based on Euclidean distance in the sequence embeddings. It can also be applied to calculate the fraction of nearest neighbors in any group.
4. Differential analysis: Conducts a Wilcoxon rank-sum test on sequence embeddings to identify features enriched or depleted relative to the reference group.

5. Classifier performance: Polygraph trains a support vector machine (SVM) classifier with *k*-fold cross-validation and assesses its ability to differentiate between each group of synthetic sequences and the reference group based on their embedding vectors. Classifier performance is measured by the area under the receiver operator curve (AUROC) metric.

Polygraph supports statistical analyses to compare any of these metrics across groups:

1. Groupwise Fisher's exact test: for comparing proportions between each non-reference group and the reference group.
2. Groupwise Mann–Whitney *U* test: for comparing mean values between each non-reference group and the reference group.
3. Kruskal–Wallis test followed by Dunn's post hoc test: for comparing mean values across all groups.

Guided evolution: The Polygraph evolve module enables custom sequence design. The evolve function performs iterative directed evolution on DNA sequences, in which all possible single-base mutations to the input sequence are explored in each iteration and the one with the best model prediction is selected for the next iteration. However, Polygraph offers users the option to simultaneously optimize for both predicted activity based on a provided model and similarity to a set of reference native DNA sequences by embedding the sequences and calculating Euclidean distances in the embedding space. The function allows the user to control the relative importance of activity versus similarity through a weighting parameter. The user can choose model-based, *k*-mer, or motif embeddings for the purpose of this calculation. The resulting sequences have high predicted activity while maintaining similarity to a native sequence population. However, since this approach balances activity and similarity, it will require more iterations to match the level of activity of sequences designed by ordinary directed evolution.

Yeast promoter dataset

Yeast gigantically parallel reporter assay (GPRA) data as well as expression measurements for native promoters were obtained from Vaishnav et al. [37]. The 50 native promoters with strongest measured activity in this dataset were denoted as Native (Strong), while the 50 with weakest measured activity were denoted as Native (Weak).

Yeast promoter design

A sequence-to-expression model was trained using a convolutional architecture on 3922 80-bp sequences of native yeast promoters whose expression was measured in complex medium. The model consisted of 4 convolutional layers each with 64 channels and ReLU activation. The first convolutional layer had kernel size $=15$ and subsequent layers had kernel size $=3$. The convolutional layers were followed by average pooling across the sequence length. Finally, a 1×1 convolutional layer was used to combine the output of the 64 channels into the predicted expression value. Three thousand five hundred

randomly selected promoters were used for training while 422 were held out each for validation and testing. The model was trained for 10 epochs with Poisson loss using the Adam optimizer with learning rate $=10^{-4}$ and batch size 16. This model was used as an oracle for sequence design.

To generate sequences using directed evolution, we took the 50 Native (Weak) sequences and iterated each one through 10 rounds of evolution using the yeast model as an oracle to increase expression. Fifty Ledidi [13] optimized sequences were generated similarly with the following parameters: max_iter $=500$, l$=$ 100, lr $=10^{-2}$. Fifty sequences were generated each through 25 iterations of the custom guided evolution function, with the Native (Weak) sequences serving as the reference population, using the pre-final layers of the oracle model as an embedding space, and alpha (weighting parameter) set to 9.

Yeast embedding model

We considered approximately 7.5 million 80-bp long randomly generated yeast promoters whose expression was measured in both complex and defined media from Vaishnav et al. [37]. One million randomly selected sequences were held out for training and validation. We trained a sequence-to-expression model to predict expression in both media. The model consisted of 4 convolutional layers with 512 channels each and ReLU activation. All convolutional layers after the first were followed by max pooling with width 2. The output of the final convolutional layer was flattened and passed through a dense layer with 32 nodes and ReLU activation to produce two outputs.

The output from the final convolutional layer of this model was used to embed native and synthetic sequences. The trained model can be downloaded and used as part of the Polygraph package.

Human enhancer dataset

Gosai et al. [15] generated a dataset of 776,474 200-nucleotide sequences paired with gene expression, as measured by a massively parallel reporter assay (MPRA) in three cell lines (HepG2, K562, and SK-N-SH). Using this model, they engineered cell-type specific candidate regulatory elements with three design approaches: simulated annealing, FastSeqProp [12], and AdaLead [14].

Here, we focused on the HepG2 cell type, for which they designed and experimentally validated a total of 16,907 synthetic regulatory elements. In addition, the authors examined a total of 8000 native regulatory elements selected on the basis of measured HepG2-specific chromatin accessibility or HepG2-specific regulatory activity predicted by their model.

We selected 16,031 synthetic regulatory elements for HepG2 which were predicted by the authors to have higher activity than the 75 th percentile native element in HepG2 cells (3.9). Sequences whose experimentally measured activity in HepG2 cells turned out to be less than the 25 th percentile native element (1.14) were denoted as having failed, while the remaining synthetic sequences were deemed successful at achieving HepG2 enhancer activity.

Human chromatin accessibility model

We finetuned a pretrained Enformer model [10] using the Enformer PyTorch implementation at https://github.com/lucidrains/enformer-pytorch to predict binarized cell type-specific pseudobulk accessibility based on the single-cell atlas of chromatin accessibility in the human genome [53]. Sequences from chromosomes 7 and 13 were held out for validation and testing, respectively. We retained only the first transformer block of the model and dropped the remaining blocks, used "target_length $=-1$" argument to disable cropping, and added a 1D average pooling to collapse the sequence dimensionality. The input sequence length was 200 bp. Binary chromatin accessibility signal and cell type annotations were obtained from publicly available files provided by the authors (http://catlas.org/catlas_downloads/humantissues/cCRE_by_cell_type/). Cell types with less than 3% positive labels were discarded which reduced the number of cell types from 222 to 203.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13059-025-03584-9.

Additional file 1: Tables S1–S9 and Figures S1–S11 with legends

Additional file 2: Peer review history

Acknowledgements
We thank David Garfield and Oriol Fornes for helpful discussion and feedback.

Peer review information
Tim Sands was the primary editor of this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Authors' contributions
AG, AL, and GE formulated the project. AL developed the polygraph codebase, with support from AG and LG. AL wrote code tutorials, with support from LG. AL integrated and trained models, with support from GE. LG performed analysis on the human and yeast datasets, with support from AL and GE. AL and LG drafted the figures and manuscript. TB provided supervision and mentorship. All authors revised the manuscript.

Funding
The authors declare no funding sources.

Data availability
Project name: Polygraph.
Project home page: https://github.com/Genentech/polygraph [54].
Documentation: https://genentech.github.io/polygraph.
Operating system(s): Platform independent.
Programming language: Python.
Other requirements: Polygraph requires Python > = v3.8. A GPU is not required but supports faster model-based embeddings and predictions.
License: MIT.
Model weights, code used in this manuscript, all sequences, and results of all analyses have been deposited at Zenodo: https://doi.org/10.5281/zenodo.14648912 [55].
Yeast gigantically parallel reporter assay (GPRA) data were obtained from [37]. Human enhancer sequences and MPRA measurements were obtained from [15].

Declarations

Ethics approval and consent to participate
Not applicable.

Competing interests
All authors were employees of Genentech, Inc. during this work.

Received: 18 April 2024 Accepted: 23 April 2025
Published online: 06 May 2025

References

1. Ong C-T, Corces VG. Enhancer function: new insights into the regulation of tissue-specific gene expression. Nat Rev Genet. 2011;12:283–93.
2. Maeder ML, Gersbach CA. Genome-editing technologies for gene and cell therapy. Mol Ther. 2016;24:430–46.
3. Naldini L. Gene therapy returns to centre stage. Nature. 2015;526:351–60.
4. High KA, Roncarolo MG. Gene therapy. N Engl J Med. 2019;381:455–64.
5. Dunbar CE, High KA, Joung JK, Kohn DB, Ozawa K, Sadelain M. Gene therapy comes of age. Science. 2018;359. https://doi.org/10.1126/science.aan4672
6. Au HKE, Isalan M, Mielcarek M. Gene therapy advances: a meta-analysis of AAV usage in clinical settings. Front Med. 2022;8: 809118.
7. Kelley DR, Reshef YA, Bileschi M, Belanger D, McLean CY, Snoek J. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. Genome Res. 2018;28:739–50.
8. Linder J, Srivastava D, Yuan H, Agarwal V, Kelley DR. Predicting RNA-seq coverage from DNA sequence as a unifying model of gene regulation. bioRxiv. 2023. p. 2023.08.30.555582. https://doi.org/10.1101/2023.08.30.555582
9. Lal A, Karollus A, Gunsalus L, Garfield D, Nair S, Tseng AM, et al. Decoding sequence determinants of gene expression in diverse cellular and disease states. bioRxiv. 2024. p. 2024.10.09.617507. https://doi.org/10.1101/2024.10.09.617507
10. Avsec Ž, Agarwal V, Visentin D, Ledsam JR, Grabska-Barwinska A, Taylor KR, et al. Effective gene expression prediction from sequence by integrating long-range interactions. Nat Methods. 2021;18:1196–203.
11. Minnoye L, Taskiran II, Mauduit D, Fazio M, Van Aerschot L, Hulselmans G, et al. Cross-species analysis of enhancer logic using deep learning. Genome Res. 2020;30:1815–34.
12. Linder J, Seelig G. Fast activation maximization for molecular sequence design. BMC Bioinformatics. 2021;22:510.
13. Schreiber J, Lu YY. Ledidi: designing genomic edits that induce functional activity. bioRxiv. 2020. p. 2020.05.21.109686. https://doi.org/10.1101/2020.05.21.109686
14. Sinai S, Wang R, Whatley A, Slocum S, Locane E, Kelsic ED. AdaLead: a simple and robust adaptive greedy search algorithm for sequence design. arXiv [cs.LG]. 2020. Available: http://arxiv.org/abs/2010.02141
15. Gosai SJ, Castro RI, Fuentes N, Butts JC, Mouri K, Alasoadura M, et al. Machine-guided design of cell-type-targeting cis-regulatory elements. Nature. 2024;634:1211–20.
16. Zrimec J, Fu X, Muhammad AS, Skrekas C, Jauniskis V, Speicher NK, et al. Controlling gene expression with deep generative design of regulatory DNA. Nat Commun. 2022;13:5099.
17. Jain M, Bengio E, Hernandez-Garcia A, Rector-Brooks J, Dossou BFP, Ekbote CA, et al. Biological sequence design with GFlowNets. In: Chaudhuri K, Jegelka S, Song L, Szepesvari C, Niu G, Sabato S, editors. Proceedings of the 39th International Conference on Machine Learning. PMLR; 17--23 Jul 2022. pp. 9786–9801.
18. Gupta A, Kundaje A. Targeted optimization of regulatory DNA sequences with neural editing architectures. bioRxiv. 2019. p. 714402. https://doi.org/10.1101/714402
19. Avdeyev P, Shi C, Tan Y, Dudnyk K, Zhou J. Dirichlet diffusion score model for biological sequence generation. ArXiv. 2023. Available: https://www.ncbi.nlm.nih.gov/pubmed/37292476
20. DaSilva LF, Senan S, Patel ZM, Reddy AJ, Gabbita S, Nussbaum Z, et al. DNA-Diffusion: leveraging generative models for controlling chromatin accessibility and gene expression via synthetic regulatory elements. bioRxiv. 2024. p. 2024.02.01.578352. https://doi.org/10.1101/2024.02.01.578352
21. Sarkar A, Tang Z, Zhao C, Koo PK. Designing DNA with tunable regulatory activity using discrete diffusion. bioRxiv. 2024. p. 2024.05.23.595630. https://doi.org/10.1101/2024.05.23.595630
22. Uehara M, Zhao Y, Hajiramezanali E, Scalia G, Eraslan G, Lal A, et al. Bridging model-based optimization and generative modeling via conservative fine-tuning of diffusion models. 2024. Available: http://arxiv.org/abs/2405.19673
23. Lal A, Garfield D, Biancalani T, Eraslan G. Designing realistic regulatory DNA with autoregressive language models. Genome Res. 2024;34:1411–20.
24. Li Z, Ni Y, Beardall WAV, Xia G, Das A, Stan G-B, et al. DiscDiff: latent diffusion model for DNA sequence generation. 2024. Available: http://arxiv.org/abs/2402.06079
25. Stark H, Jing B, Wang C, Corso G, Berger B, Barzilay R, et al. Dirichlet flow matching with applications to DNA sequence design. ArXiv. 2024. Available: https://www.ncbi.nlm.nih.gov/pubmed/38855543
26. Fishman N, Shrikumar A, Marinov GK, Kundaje A, Systematic characterization of generative models for de novo design of regulatory DNA. https://icml-compbio.github.io/2020/papers/WCBICML2020_paper_46.pdf
27. Colbran LL, Chen L, Capra JA. Short DNA sequence patterns accurately identify broadly active human enhancers. BMC Genomics. 2017;18:536.
28. Colbran LL, Chen L, Capra JA. Sequence characteristics distinguish transcribed enhancers from promoters and predict their breadth of activity. Genetics. 2019;211:1205–17.
29. Horton CA, Alexandari AM, Hayes MGB, Marklund E, Schaepe JM, Aditham AK, et al. Short tandem repeats bind transcription factors to tune eukaryotic gene expression. Science. 2023;381:eadd1250.
30. McInnes L, Healy J, Melville J. UMAP: uniform manifold approximation and projection for dimension reduction. arXiv [stat.ML]. 2018. Available: http://arxiv.org/abs/1802.03426
31. Hie B, Cho H, DeMeo B, Bryson B, Berger B. Geometric sketching compactly summarizes the single-cell transcriptomic landscape. Cels. 2019;8:483-493.e7.
32. Nguyen E, Poli M, Faizi M, Thomas A, Birch-Sykes C, Wornow M, et al. HyenaDNA: long-range genomic sequence modeling at single nucleotide resolution. ArXiv. 2023. Available: https://www.ncbi.nlm.nih.gov/pubmed/37426456
33. Taskiran II, Spanier KI, Christiaens V, Mauduit D, Aerts S. Cell type directed design of synthetic enhancers. bioRxiv. 2022. p. 2022.07.26.501466. https://doi.org/10.1101/2022.07.26.501466
34. van Laarhoven PJ, Aarts EH. Simulated annealing: theory and applications. Springer Science & Business Media; 2013.
35. Taskiran II, Spanier KI, Dickmänken H, Kempynck N, Pančíková A, Ekşi EC, et al. Cell-type-directed design of synthetic enhancers. Nature. 2024;626:212–20.
36. de Boer CG, Vaishnav ED, Sadeh R, Abeyta EL, Friedman N, Regev A. Author correction: deciphering eukaryotic gene-regulatory logic with 100 million random promoters. Nat Biotechnol. 2020;38:1211.

37. Vaishnav ED, de Boer CG, Molinet J, Yassour M, Fan L, Adiconis X, et al. The evolution, evolvability and engineering of gene regulatory DNA. Nature. 2022;603:455–63.
38. Wood V, Gwilliam R, Rajandream MA, Lyne M, Lyne R, Stewart A, et al. The genome sequence of Schizosaccharomyces pombe. Nature. 2002;415. https://doi.org/10.1038/nature724
39. Rauluseviciute I, Riudavets-Puig R, Blanc-Mathieu R, Castro-Mondragon JA, Ferenc K, Kumar V, et al. JASPAR 2024: 20th anniversary of the open-access database of transcription factor binding profiles. Nucleic Acids Res. 2024;52:D174–82.
40. Nasmyth K, Dirick L. The role of SWI4 and SWI6 in the activity of G1 cyclins in yeast. Cell. 1991;66:995–1013.
41. Boonstra J. Regulation of G1 phase progression. Springer Science & Business Media; 2003.
42. Bricmont PA, Daugherty JR, Cooper TG. The DAL81 gene product is required for induced expression of two differently regulated nitrogen catabolic genes in Saccharomyces cerevisiae. Mol Cell Biol. 1991;11:1161–6.
43. Holmberg S, Schjerling P. Cha4p of Saccharomyces cerevisiae activates transcription via serine/threonine response elements. Genetics. 1996;144:467–78.
44. Siddiqui AH, Brandriss MC. The Saccharomyces cerevisiae PUT3 activator protein associates with proline-specific upstream activation sequences. Mol Cell Biol. 1989;9:4706–12.
45. Gray WM, Fassler JS. Isolation and analysis of the yeast TEA1 gene, which encodes a zinc cluster Ty enhancer-binding protein. Mol Cell Biol. 1996;16:347–58.
46. Floer M, Wang X, Prabhu V, Berrozpe G, Narayan S, Spagna D, et al. A RSC/nucleosome complex determines chromatin architecture and facilitates activator binding. Cell. 2010;141:407.
47. Kiktev DA, Sheng Z, Lobachev KS, Petes TD. GC content elevates mutation and recombination rates in the yeast Saccharomyces cerevisiae. Proc Natl Acad Sci. 2018;115:E7109–18.
48. Cereghini S. Liver-enriched transcription factors and hepatocyte differentiation. FASEB J. 1996;10:267–82.
49. Stepniak E, Ricci R, Eferl R, Sumara G, Sumara I, Rath M, et al. c-Jun/AP-1 controls liver regeneration by repressing p53/p21 and p38 MAPK activity. Genes Dev. 2006;20:2306.
50. Grant CE, Bailey TL, Noble WS. FIMO: scanning for occurrences of a given motif. Bioinformatics. 2011;27:1017–8.
51. Lonsdale J, Thomas J, Salvatore M, Phillips R, Lo E, Shad S, et al. The Genotype-Tissue Expression (GTEx) project. Nat Genet. 2013;45:580–5.
52. Dalla-Torre H, Gonzalez L, Mendoza-Revilla J, Carranza NL, Grzywaczewski AH, Oteri F, et al. The Nucleotide Transformer: building and evaluating robust foundation models for human genomics. bioRxiv. 2023. p. 2023.01.11.523679. https://doi.org/10.1101/2023.01.11.523679
53. Zhang K, Hocker JD, Miller M, Hou X, Chiou J, Poirion OB, et al. A single-cell atlas of chromatin accessibility in the human genome. Cell. 2021;184:5985-6001.e19.
54. Lal A, Gunsalus L. Genentech/polygraph: v0.1. Zenodo; 2025. https://doi.org/10.5281/ZENODO.15098651
55. Lal A. Polygraph: a software framework for the systematic assessment of synthetic regulatory DNA elements. 2025. Zenodo. https://doi.org/10.5281/zenodo.14648912.

Publisher's Note