Synthetic Promoter Design in *Escherichia coli* based on Generative Adversarial Network

Ye Wang[1, a], Haochen Wang[1, a], Liyang Liu[1], Xiaowo Wang[1, *]

AUTHOR ADDRESS: [1]Ministry of Education Key Laboratory of Bioinformatics; Center for Synthetic and Systems Biology; Bioinformatics Division, Beijing National Research Center for Information Science and Technology; Department of Automation, Tsinghua University, Beijing, 100084, China

[a] These authors contribute equally

[*] Correspondence: xwwang@tsinghua.edu.cn

KEYWORDS: synthetic promoters, generative adversarial network, transcriptional fine-tuning, computational design.

ABSTRACT: Synthetic promoters are commonly applied elements in circuit design for fine-tuning the protein expression levels. Promoter engineering was mostly focused on the random mutation or combination of regulation elements such as transcription factor binding sites. However, the size of promoter sequence space is still overwhelming and better navigation method is required. On the other hand, the generative adversarial network (GAN) is known for its great ability to reduce the searching space by learning to generate new data on the similar manifold of original data. Here, we applied WGAN-GP model into *de novo* promoter sequence design to generate entirely new promoter sequences. In total, 83 of model-generated promoter sequences were tested in promoter activity screening by regulating the expression of *sfGFP* gene in *Escherichia coli*. As a result, 26 out of 83 newly designed promoters were found functional and successfully expressed with varying activities, with similarity score to natural promoters all less than 0.7. Moreover, 3 of them showed higher promoter strength than the wild type promoters and their highly expression mutants. The much higher successful rate and promoter activity with much lower similarity score in our model-designed novel promoters confirmed the effectiveness of promoter sequence learning. Our work provides insights into an area of navigation of novel functional promoter sequence space automatically, as well as speeding up evolution process of naturally existing promoters, indicating the potential ability for deep generative models to be applied into genetic element designing in the future.

INTRODUCTION

Promoters are cis-regulatory DNA elements that recruit transcription factors and initiate transcription process. Although natural promoters have been widely used in relatively steady laboratory application, the number of functional promoters, however, is in such a low level, considering the large potential space of base combinations. In recent years, researchers increasingly turned to extract promoter features[1-3] and constructed libraries of synthetic promoter sequences for specific engineering aims[4-6]. Many molecular approaches to generate of synthetic promoter libraries has been applied, such as methods based on mutagenesis [6-8], integrating the shorter functional parts [9-11] and combination of biological motifs [12, 13], etc.

Molecular methods based on constructing random mutation library, such as error-prone PCR, was reported to successfully mutagenize a bacteriophage PL-λ promoter placed upstream of a green fluorescent protein (GFP) sequence in *E.coli*, resulting in a library with approximately 9000–12000 functional clones [14, 15]. Other methods based on inserting non-consensus gaps [9] into the biological motifs [16] also provided some novel synthetic promoters for fine-tuning the target genes.

However, one of the limitations of the molecular methodologies may be that, although above methods have certainly provided new promoters with a range of different expression levels, these approaches still do not provide a systematic, theoretical navigation of the promoter design space [17]. Besides, these newly designed promoters usually shared a relatively high similarity in sequence, making the library diversity being restricted. Hence, a number of computational approaches have been available for *de novo* promoter sequence design[18].

The position weight matrices (PWMs) models, for example, has been proved to show some promise in detecting transcription factor binding sites (TFBS) and prediction of promoter strength. [19, 20]. A relatively high degree of accuracy was obtained in predicting promoter strength regulated by *E. coli* σE by encompassing both upstream sequences and core promoter elements into the PSSL model

[21]. However, such approaches could be inadequate to model poorly conserved house-keeping promoters, such as those regulated by E. coli σ70, for the greater complexity required within these models.

Meanwhile, thermodynamic models analyzed the role of transcription factor binding sites in gene expression process, by considering the binding energy of RNA polymerase [22]. The predicting expression for the simple repression strains could be within 30% after correcting the prediction using the data from the unregulated promoters [23]. However, the number of unique binding sites of promoters were still limited and a larger screening space for energy matrix may also be needed.

For typical prokaryotic organisms, if only considering promoter sequences within 50 bp in length, the number of potential promoter sequences is estimated to be on the order of 4^{50}. Similar estimation could be much larger for eukaryotes due to the genome complexity. Therefore, any global strategy for covering this space might seem impossible, and better navigation method is needed.

The generative models provide a better way of probing large data space to generate entirely new data. Among them, the generative adversarial networks (GAN) have shown its great ability in mimicking the target distribution of real data set to generate new data which is on the similar manifold of real data [24]. Based on the game theory between the generator and discriminator, GAN has given birth to sets of state-of-art image generation methods and showed its ability to generate completely new images. Indeed, different variants of GAN models have increasingly been used to generate entirely new biological data, such as brain activity reconstruction images [25], synthetic DNA sequences encoding for proteins of variable length [26] and single cell RNA-seq data from multiple cell types [27].

Here, we proposed the first approach for *de novo* promoter sequence design method based on a generative adversarial network (GAN) and validated the strength of generated promoters in *Escherichia coli*. Taking the natural promoter sequences as an input, the generator learned to produce novel promoter sequences that the discriminator cannot classify from the natural ones. The novel promoter sequences were derived from WGAN-GP model firstly, with another model of deep Convolutional Neural Network (CNN) was used to select generated promoters with the highest expected expression levels. Using the deep generative network, we have created an artificial library with 83 novel promoter sequences, and their relative promoter activities were all demonstrated *in vivo*.

RESULTS

Framework description of promoter designing based on Generative Adversarial Network. For the aim of generating functional synthetic promoters, we introduced three main components in our whole method framework [figure.1], including promoter generation, high strength promoter selection and the experiment validation.

The Promoter generation stage provided thousands of synthetic promoters using the generative adversarial network (GAN). The training set data was the experimentally identified naturally occurring promoters in *E.coli* obtained from *E.coli* database RegulonDB[28]. Specifically, only the promoter sequences recognized using the dRNA-seq technic[29] published by Thomason et al[30]. were used in present work. The training set contains 14098 promoters and most of them are sigma 70 promoters.

GAN model has two main adversarial components, the discriminator and generator. The basic workflow of GAN goes as follows: Taking the naturally occurring promoters as inputs, the discriminator evaluates the divergence between the current generated promoters and the natural promoters. The generator generates entirely new promoter sequences by dealing with random noise. By training the discriminator and generator in turn iteratively, the discriminator gradually learns to act as an increasingly meticulous critic of the promoters generated by current generator. The generator, on the contrary, learns to improve its skills of mimicking the naturally occurring promoter distribution in order to the deceive discriminator. By producing promoters from final trained generator, GAN could help us navigate the possible promoter space nearby the input natural ones to generate new synthetic promoters.

Specifically, our method was mainly based on the WGAN-GP framework (material and methods), which was concentrated on generating 50bp length promoters at the upstream of TSS, whereas 5'UTR region was fixed and also acted as an insulator sequence to avoid interaction effects between core transcriptional elements[16].

Promoter selection stage used a convolutional neural network (CNN) based model to predict the gene expression level of the model-generated promoters. The model structure is shown in [figure.3]. Only the promoters with top predicted promoter strength were remained for experimental validation.

In the experimental validation stage, the selected synthetic promoter sequences were used to build the promoter library in *E.coli* with a fixed 5'UTR region, driving the expression of *sfgfp* gene and verify the expression levels *in vivo*.

The architecture of WGAN-GP for promoter generation. At first, we supposed the DCGAN based network structure could fit in well with synthetic promoter generation. Ideally, convolutional layer could grasp motif feature and multilayer network could learn the abstract representation of promoter regulation rules[31]. The detail description about the network is illustrated in "materials and methods".

We tested the motif feature learning of DCGAN model *in silico*. The comparison of sequence logo is shown in [figure.2(b)] by weblogo[32], which demonstrates that –10 and –35 region logo could be partially learned by our model. However, it is obvious that DCGAN model suffered several problems in motif feature learning. The model overestimates the importance of several motifs, meanwhile part of

some features are ignored[figure.7]. Besides, the model generation results are instable and often falls into local optimal solution [figure.2(b)]. The DCGAN based model is not completely met with our case. Thus, we tried several sequence-based generation models, and finally WGAN-GP language model was selected in order to handle the above problems.

The WGAN-GP model was originally designed for the nature language generation task. The whole structure is shown in [figure.4(a)]. Two main changes were used to solve problems in DCGAN: the introducing of resblock structure[33] and the utilizing of earth moving distance.

Resblock is combination of convolutional layers, and it adds a short connection from resblock start to resblock end. One of the resblock structures are shown in [figure.4(b)]. Resblock structure was proposed by He et al to handle the degradation problem, and the network would have deeper layers with stronger feature learning ability. In our work, we found that in the synthetic promoter generation, resblock structure could decrease the local minimum problem compared with the same depth neural network without short connections [figure.5]. The final promoter generation results also verifies the success of resblock structure: the local minimum problem no longer exists by surveying the sequence logo.

Besides, the WGAN-GP language model uses earth-moving distance instead of JS divergence[34] to measure the distance between the probability distribution of true samples and generated samples. Earth moving distance is continuous and differentiable in the whole space[35], so that we could progressively train the model until the generated samples manifold converge to true samples manifold.

In our work, such property means the improvement of stability and feature learning ability of the model. The generation results show that overestimation and underestimation are not appeared in the WGAN-GP model. [Figure.7]. And the -10 and -35 region motif logo was nearly perfect learned by the model. [figure.5]

The promoter activities generated by GAN models and similarity score of sequences. In total, 83 synthetic promoters from WGAN-GP model selected by mentioned prediction model were tested in *Escherichia coli*. The detailed sequences are provided as Supplementary Table S1. In the 83 novel promoters, 31.3% (26 out of 83) designed promoter sequences were demonstrated to have higher activity than control plasmids and 11.5% of them showed higher promoter strength than the highest member of six positive control promoters (figure.6). Three groups of promoter sequences with not functional, functional but not higher and higher labels were analyzed. The promoters with higher strength showed a more than 1.2-fold relative activity stronger than the average of six positive control promoters, and the functional but not higher group showed a comparative expression level with the wild-type controls.

We checked the similarity score between the natural promoter group and the generation promoter group by using CLUSTALW2[36]. Specifically, we compared our 83 synthetic promoters with the training promoter groups[30], the minimum of the max scores is 54, the maximum is 72, and the mean is 62.04. These results verify synthetic promoters show no significant matches to the training promoters. A standard nucleotide BLAST searching was also conducted against the whole *E.coli* K12 genome and no significant matches were found. The highest e-value obtained for candidate promoter sequences was 0.014, which is a much higher than the mutation libraries, indicating the newly designed promoter sequences is different from the naturally occurring sequences. Such feature could increase the genetic stability in the genome context, because of the low probability to recombine with sequences in the *E.coli* genome.[9]

The high ratio of functional promoters and low level of similarity score in our novel 83 promoter library confirmed the effectiveness of distribution learning of naturally occurring promoters, indicating the discovery of novel functional promoter space in *E.coli*.

Characteristics of generated promoters. The kmer frequency and the space correlation of motifs were also tested for the characteristics of newly generated promoter sequences.

The kmer frequency distribution compared the occurrence number of certain kmer between natural promoter groups and generation promoter groups. Five of top ten 6-mers were the same, which indicated that our WGAN-GP model could learn the most important features from the natural promoter group. For the whole view of kmer learning ability, kmer frequency from k = 2 to k = 6 was plotted. The inconsistent frequency of certain kmer between the synthetic group and natural group indicated that the learning model may overestimate or underestimate the importance of a certain kmer, and the consistent frequency of a certain kmer was a good feature of the synthetic promoter groups. We also used pearson correlation coefficient (r-square) to measure the correlation between the natural promoters and synthetic promoters. As shown in [figure.7], the kmer feature of synthetic promoter learned by WGAN-GP could correlate well with the natural promoter.

Besides, the location distribution of five most frequent 6-mer was also checked, to inspect whether our model could learn the right location of certain kmer on the promoter. The results show the WGAN-GP model could not only learn the right frequency of kmer, but also learn the right location of a certain kmer. That is to say, our synthetic promoters conform to the kmer feature of the natural promoters.

In addition, the space correlation of motifs was also compared by the distance distribution of -10 motif and -35 motif. Previous study has shown that the distance between -10 region and -35 region with 16 ~ 18 bp is proper for a promoter sequence[37], which could be beneficial to the binding of RNA polymerase. The distance of our generation promoters concentrated on the area between 16 and 18, which is consistent with the distance of natural promoters[figure.8].

We also compared our synthetic promoter generation results with the consensus sequence sampling results. We counted the occurrence possibility of each position of promoter sequences, and generated the promoter sequences by such occurrence possibility. We firstly compared the kmer frequency plot between the WGAN-GP generation promoters and the consensus sequence sampling promoters. As shown in [figure.7], compared with WGAN-GP promoters, the consensus sequence sampling promoters are not matching well with the frequency distribution of natural promoters. This could also be reflected by the R-squared criterion: the relation between the kmer frequency of consensus sequence sampling promoters and the natural promoters drop quickly with the growing number of k. [figure.7] Besides, the locating distribution of most frequent five 6-mer illustrates the WGAN-GP generation promoters consistent better with the natural promoters[figure.8(a)] Similarly, the space correlation of WGAN-GP promoter groups is better than the consensus sequence sampling promoter group. [figure.8(c)]

DISCUSSION

Promoter region is one of the most fundamental tuning elements driving the transcription of a given gene by influencing the binding of RNA polymerase in both prokaryotes and eukaryotes. However, the number of functional naturally occurring promoters is limited, compared with the huge possible space of base combination. Synthetic promoters could optimize the natural promoters, then initiate the comparatively higher gene expression level than naturally existing promoters for specific tasks. Traditional molecular and computational methods each for novel functional promoter sequences mainly centered on random mutation and artificially combine the motifs of promoter regions. We applied the generative adversarial network into the de novo promoter designing, and our result shows a higher ratio of functional promoters with better feature extraction in promoter sequences.

Naturally occurring functional promoters have evolved for billions of years, however, by taking advantage of manifold learning ability of GAN models, we could generate thousands of candidate promoter sequences in silico conveniently. We mainly build our generative model based on WGAN-GP, then the prediction model was used to select promoters. The experimental results also showed a high ratio of 46% of 50 promoter sequences are found functional in E.coli. Higher order sequence features and similarity analyzing also shows that the generated promoters not only inherent the important motifs from the natural promoter population, but also differ greatly from the natural promoters. Our work indicates the potential application of generative models for genetic element designing tasks.

By an evolutionary perspective, the two adversarial components in GAN model, say the generator and discriminator, play similar roles as the naturally excessive reproduced organisms and environmental selection pressures. The generator transfers the noise z, which is similar to the permutation from environment to organisms, step by step finally to the layer of new DNA sequence. The discriminator, on the other hand, could play a role as natural selection pressure to score the newly produced promoters. At the initial stage, the discriminator converges quickly to form a relatively steady environment. Then the promoter fitness score gradient is backpropagated to the generator, helping it to correct its parameters and learns to generate newly evolved promoters with higher fitness scores. This process has been performed iteratively for many epochs in turn, and the final novel promoters with the highest fitness scores given by trained discriminator was used in the present work. Unlike the naturally occurring organism, which generally mutant randomly and passively, our generator network could correct its parameters directionally, which indeed speeds up the evolution process, but also come up with less robustness in the whole model system. This phenomenon has also been reported as a common challenge for training GAN models in different fields.

Moreover, some other variants of generative adversarial network may also be introduced into element designing: the conditional GAN model (cGAN)[38], for example, could be implemented into designing of inducible promoters, which conditions on the types of inducers in the different biological occasions.

In conclusion, we applied the generative adversarial network into the promoter sequence design and experimentally validated the promoter activities in vivo. Our work provides insights into an area of artificially speeding up evolution of naturally existing promoters, and navigation of novel functional promoter sequence space, indicating the potential ability for generative networks to be used in elements designing in the future.

MATERIALS AND METHODS

Strains and Cultures. The *E.coli* strain DH5α[F^-, λ^-, $\varphi 80 lacZ\Delta M15$, $\Delta(lacZYA-argF)$, U169, endA1, recA1, hsdR17(rk^-, mk^+) sup E44, thi^-, gyrA96, relA1, phoA] was used as host organism and cultivated in Luria–Bertani (LB) media supplemented with 100 μg/mL kanamycin at 37 °C for promoter activity screening.

Promoter library plasmid construction. All promoter constructs were carried on the high-copy-number vector T1.3, driving the expression of sfGFP from the reporter gene *sfgfp*. Promoter and 5'UTR sequences are displayed in Table S1 of the supplemental material. The 5'UTR fragments (B0030) contained an EcoRI site and reverse primers contained an XbaI site. The putative Shine-Dalgarno sequence in 5'UTR was placed 7nt upward of the original TSS region. Annealing reactions were performed by incubating the complementary oligonucleotides at 95 °C for 3 min (5 μl of 100 μM forward and reverse oligonucleotides in sterile water) followed by 95 °C for 1 min each cycle

with 57 cycles and cooling to 4°C for storage. The phosphorylation of annealed oligonucleotides was performed using T4 polynucleotide kinase (PNK from NEB) with ATP for 1 hour. Then the 5'UTR oligonucleotides were cloned into the EcoRI-XbaI sites in T1.3 constructs by T4 ligase catalyzing reaction. The recombinant plasmids with 5'UTR sequences were verified by sequencing. The adenine instead of thymine were designed in 5'UTR oligonucleotides at XbaI site, therefore the original XbaI site could not be digested in following steps. The T1.3 constructs with 5'UTR oligonucleotides were digested with restriction enzymes EcoRI and XbaI, and designed promoter oligonucleotides flanked with the same MCS were cloned into the new EcoRI-XbaI sites. Six positive controls promoters, three random baseline promoters and one blank control plasmid were also tested for promoter activity screening. All of the reporter plasmids were verified by sequencing.

In vivo promoter assays. Assay strains were stored as glycerol stocks in deep 96-well plates (1 ml). Cultures were grown in plates containing 5ml LB medium with kanamycin, inoculating from glycerol stocks using a sterilized metal pinner. Monoclonal selections were grown overnight (16 h) in 96-well U-bottom deep-well plate covered with sterile breathable sealing film (sterile sealing films; Axygen) at 30°C with shaking at 850 rpm on obrital shaker. The following day the overnight cultures were diluted 1:100 into a final volume of 1.8ml of fresh medium with appropriate kanamycin. In total 200ul cultures were added in clear bottom black plates, and repeated measurements of the optical density at 600 nm (OD600) and fluorescence (relative fluorescence units [RFU]; excitation at 485 nm and emission at 510 nm) were performed by microplate reader-incubator-shaker (Thermo) every 1 hour during 2 to 6 hours after culture dilution. All experiments were repeated at least three times, with less than 5% standard error between replicate experiments, showing the difference between batch cultures and microplate reader plates were negligible. Changing the well positions of strains were also conducted in the microplates during three repeated experiments to avoid any local position effects.

Establishing a Baseline Expression Level of Functional Promoters. Three control variants were generated in which the synthetic promoters were replaced by complete random sequences at each base position, with the GC content was controlled at 50%. The resulting expression levels measure the basal expression of the *sfgfp* gene given that there is enough sequence space upstream of the protein coding sequence for RNA polymerase to bind. This experiment was conducted for testing the basic background transcription in our system. The average relative promoter activity of the three control variants was 5.9% of the wild type promoter fluorescence (Figure.5). This value has been used as threshold to evaluate whether the synthetic core promoters are different from random sequences.

Promoter system designing under control promoters. Six positive control promoter sequences were used in the present work, including two different types of wild-type promoters: J23119 and Ptrc, with two of their corresponding mutants (J23100, J23102 and Ptrc_m010, Ptrc_m004), which have shown highest expression in previously reported mutation library.

A blank control variant was designed by replacing the synthetic promoter sequences with a 10bp random sequence (GGGCTCTGTA), which could not provide enough length for RNA polymerase to bind on the upward sequence of protein coding region. We performed the blank control reporter to avoid the background noise in the system and this value was also used for relative promoter activity calculating.

Promoter strength analysis. Culture background fluorescence was determined using a blank control variant plasmid carrying the same 5'UTR sequence without promoter sequence. This control T1.3 vector strain was grown and assayed under the same condition with the promoter library. The fluorescence of tested promoters subtracts background fluorescence from the reporter strain fluorescence under OD for each reporter vector, which was calculated as

$$\textit{Relative Peromoter activity} = \frac{\left(\frac{F}{OD_{600}}\right)_{Clone} - \left(\frac{F}{OD_{600}}\right)_{blank}}{\left(\frac{F}{OD_{600}}\right)_{BBa_J23119} - \left(\frac{F}{OD_{600}}\right)_{blank}}$$

The growth rate was also measured by average optical density at 600 nm (OD600) versus time(h-1). Final reported promoter activities are calculated by taking the average of three independent biological experiments.

The database description. The training dataset contains in total 14098 experimentally identified promoters in the *E.coli* K12 MG1655 genome. As the description of the dataset, most of the promoters recognized in this dataset were sigma70 promoters.

The length of these promoter sequences is 50bp, which locate at the upstream of transcription start site (TSS). The promoter sequences of 50bp length is reasonable for the -10 and -35 regions are contained in such length sequences. Meanwhile, the DNase I footprint of RNA polymerase on an *E.coli* promoter extends from approximately -50 to +20 of TSS.

The basic introduction of GAN. Generative adversarial network (GAN) has achieved impressive results in the fields of nature image generation[39, 40], style transformation[41], super resolution image creation[42]. Recent studies show that by learning the high dimensional representation of real sample group, GAN model has the ability to generating thousands of completely new samples. More specifically, GAN model contains two parts, the generator and the

discriminator. The aim of generator is to generate fake samples that could not be distinguished with real samples by the discriminator, whereas the aim of discriminator is learning to classify the fake samples and the real samples. With iteratively optimizing two counter objectives, the generator could progressively generate fake samples just as real samples.

Mathematically, in the original version of GAN[43], the aim of discriminator is as follows:

$$\max_{D} V(D,G) = E_{x\sim p_{data}(x)}\left[\log\left(D(x)\right)\right] + E_{z\sim p_z(z)}\left[\log\left(1-D\left(G(z)\right)\right)\right]$$

Here, D(x) represents the classification result of real samples. z represents input of generator, which usually is the low dimension random noise. G(z) represents generated fake samples, and D(G(z)) represent the classification result of fake samples. The first part demonstrates the discriminator should recognize real samples better, and the second part means it should recognize generated samples better. On the contrary, the aim of generator is:

$$\min_{G} V(G) = E_{z\sim p_z(z)}[\log\left(1-D\left(G(z)\right)\right)]$$

It means the generator wants to generate images that has the high possibility to fool the discriminator. Thus, GAN model is playing a minmax game by the generator and discriminator, which could be shown in the value function:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_{data}(x)}\left[\log\left(D(x)\right)\right] + E_{z\sim p_z(z)}\left[\log\left(1-D\left(G(z)\right)\right)\right] \text{(i)}$$

WGAN-GP language model. The equation (i) implies that the model is minimizing the JS divergence between the real and fake samples distribution when the discriminator is optimal[43]. However, when two distribution have supports are disjoint or lie on low dimension manifolds, such JS divergence could not iteratively make the model generate the correct samples[34]. Thus, the author used earth moving distance instead of JS divergence[35] to measure the distance between two distributions, and such GAN model is called WGAN. Then, Gulrajani I et al. add a gradient penalty into WGAN value function to solve the optimization difficulties in the original WGAN[44]. The WGAN-GP models have recovered the instability of original GAN model and achieved great results in the image and nature language generation. In our work, we use WGAN-GP language model, which uses the above technique to generate nature language, to our promoter generation task. Such model has been used in the DNA sequence generation[45], which repeatedly verifies the effectiveness of WGAN-GP model.

Manifold explanation of GAN. In the GAN structure, notice that the input of generator is comparatively lower dimension random noise, and the output of generator is comparatively higher dimension samples. Such design can represent the samples distribution confined to a low dimensional manifold[35]. Because of the exponential growth of searching space, researchers are incapable to navigate the whole space to find the optimal solution. Thus, navigating a low dimension manifold to find a feasible solution is a reasonable way in the designing work. For example, the molecular inverse design uses GAN model to navigate the manifold of functionality of chemical space to find the new materials.[17] In our work, we introduce GAN to the synthetic promoter design, in order to find the possible high expression promoter manifold.

Find the -10 and -35 motif by PSSM matrix. We first counted the occurrence possibility of each position of promoter sequences to calculate PWM matrix, then used logit function, i.e. log p_ij/b_i , where p_ij implies the element in the PWM matrix and b_i implies the background distribution. Here, we selected background by calculating the occurrence possibility of TCGA in the whole dataset. The threshold that recognize the motif was set to 3, meanwhile the -10 and -35 motif finding region was restricted to -1 ~ -17 and -40 ~ -25.

Top 6-mer distribution in -10 region. We drew eighteen top occurrence number 6-mer in -10 region. Notice that when the nucleotide C and G appear in a certain 6-mer, the PWM sampling method has more chances not to catch the distribution. [figure.9]

Several important parts in the GAN network

1. Convolution layers: The convolution layer is the core part in the promoter feature learning. It contains a great deal of convolution kernels, and each kernel could learn a certain kind of feature, which could also be interpreted as motif. The parameters of each kernel could be learned during the iteratively training of GAN model. With the combination of several Convolution layers, the whole neuron network could get the information of the promoter sequences in different sizes. The utilization of convolution layer could guide the generator to produce the certain motif in the generated promoters, and guide the discriminator to better classify the samples.

2. Upsampling layers: The aim of generator is to learn a certain data distribution. The input of the generator is a low dimension vector whereas the output is a relatively higher dimensional vector. Thus, we need upsampling layers to accomplish the mapping from the input to the output.

3. Full connection layer: This kind of layer is the most common layer in the neuron networks, it could achieve linear mapping from one vector to another vector. The convolution layer could be treated as the optimization version of full connection layer, and we use convolution layer more common than the full connection layer.

4. Max pooling layer: The max pooling layer could reduce the dimension of the feature map which extracted by the convolution layer. The multiple combination of convolution layers and max pooling layers could extract the long ranges feature in the samples.

5. Activation function: The activation function could dramatically influence the generated results. The bad choice of activation function may induce to mode collapse problem. In our first version model, we introduce tanh as the activation function of the intermediate layer. And we use the softmax as the activation function of output layer of generator. We use softmax rather than tanh as activation function because the appearance of A means the absence of T, C and G, which means the four bases are mutually exclusive. The use of softmax function will make the generator tend to generate one certain base in one location. For the output layer of discriminator, we use sigmoid as the activation function.

Encoding mode of promoter sequence Promoter sequences with length N was represented as matrix, where the length N equaled to 50. In such encode system, the T was encoded as [1,-1,-1,-1], A was encoded as [-1,1,-1,-1], and C or G was encoded in the same way. We used such code encoded by nature promoters as the real samples in the GAN model learning.

Model training of GAN models and CNN prediction model We used all the promoter sequences in the dataset described above as the real samples of GAN model. In the DCGAN model, the input of the generator was the uniform distribution random noise. The batch size was set to 128, the iteration times was set to 100 and we used stochastic gradient descent as the optimization method of our model.

Different from the DCGAN, in the WGAN-GP model, we sampled from standard normal distribution as the input noise of generator. The batch size was set to 32 and we trained our network for 160 epochs. We found that the best result is around 12 epochs, so we selected 320 synthetic promoters from that range of iterations. Notice that for giving the best play of the WGAN-GP model, we train 5 times discriminator and one times generator in each batch training. The optimizer used Adam with learning rate equals to 0.0001, beta1 equals to 0.5 and beta2 equals to 0.9.

In the training of CNN model, the dataset is from the Thomason et al. They use RNA-seq reads number as the gene expression level of a certain promoter. Thus, we have 14098 promoters with corresponding gene expression level. The batch size was set to 128, and we used stochastic gradient descent as the optimization method. We trained this model with 9000 samples as training set, 1000 samples as validation set and others as test set. Notice that we used kernel size equaled to seven in the first convolution layer, because we believe the long kernel could capture the motif like -10 motif in the promoter region.

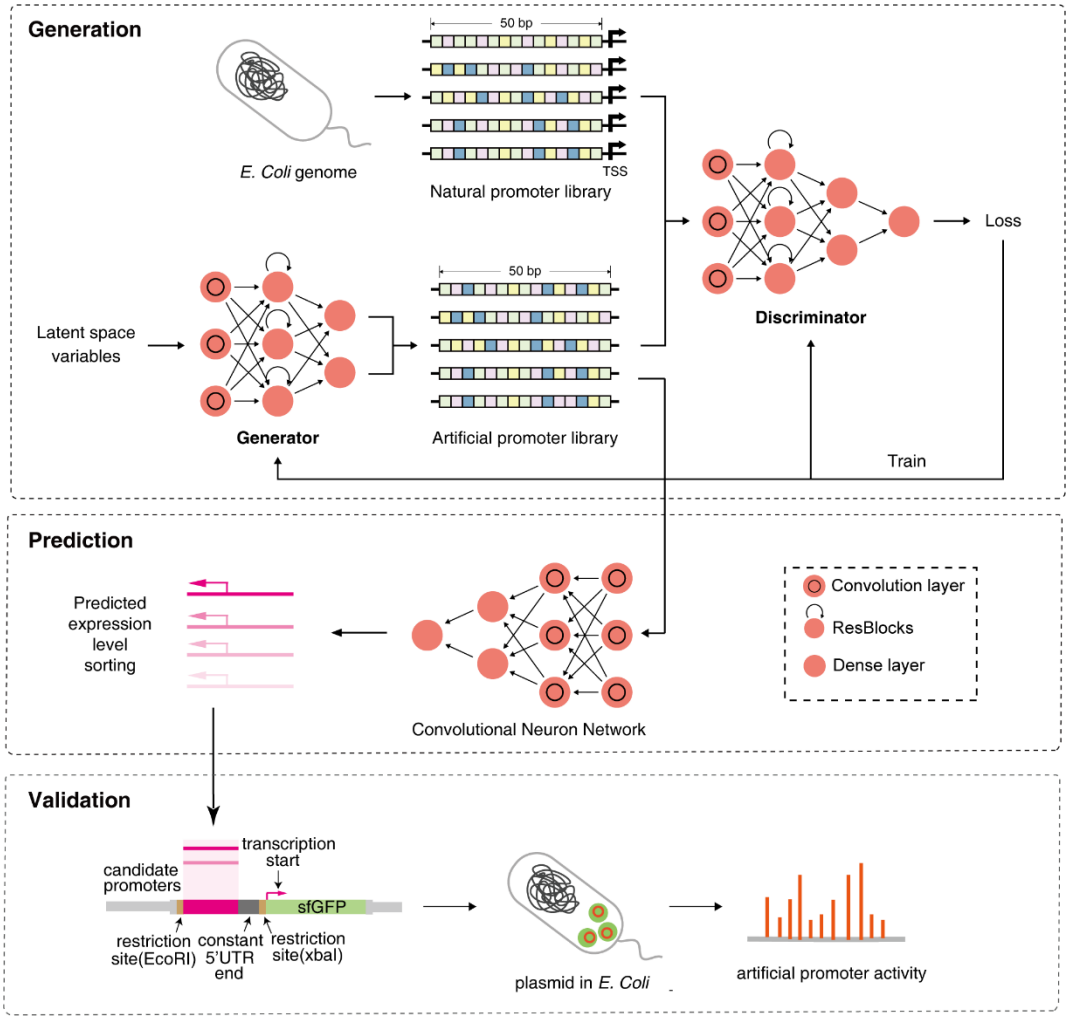
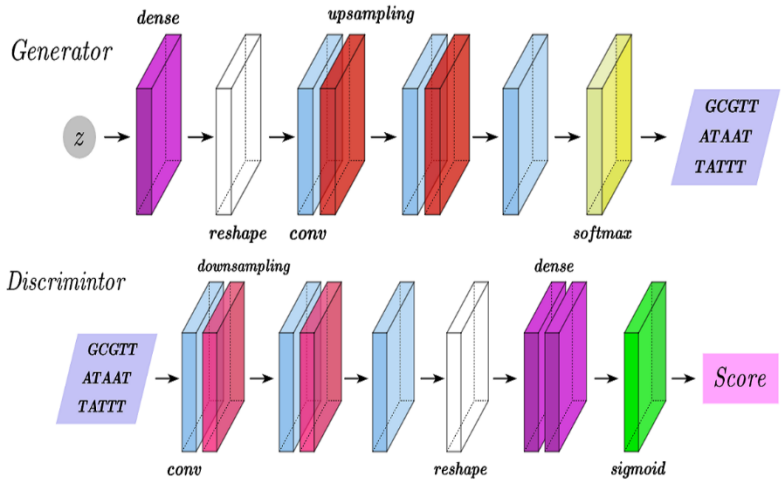
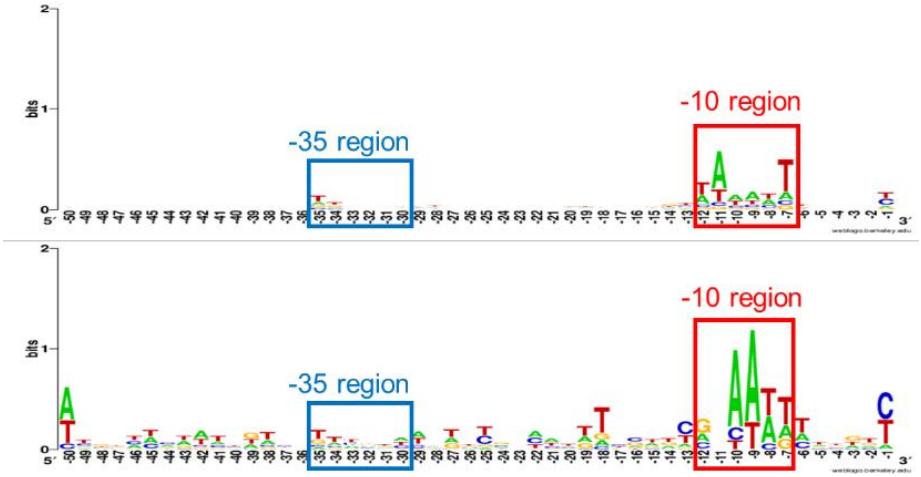


Figure.1 The synthetic promotion design system. In generation part, a large member of synthetic promoters were generated by GAN model. In selection part, the high potential expression promoters were selected by CNN model. Last, the selected promoters were verified experimentally in the validation part.



(a)



(b)

Fig.2 (a) The structure of DCGAN, the input of generator is the latent variable *z*, whereas the input of the discrimintor is one-hot code of promoter sequences. Different color represents different kinds of layer. (b) The sequence logo of -1 ~ -50 region of transcriptional start site of natural promoters(top) and DCGAN synthetic promoters(bottom). The -10 region and -35 region are annotated in the figure.

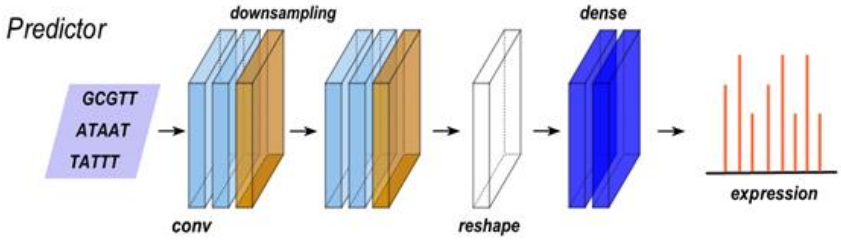
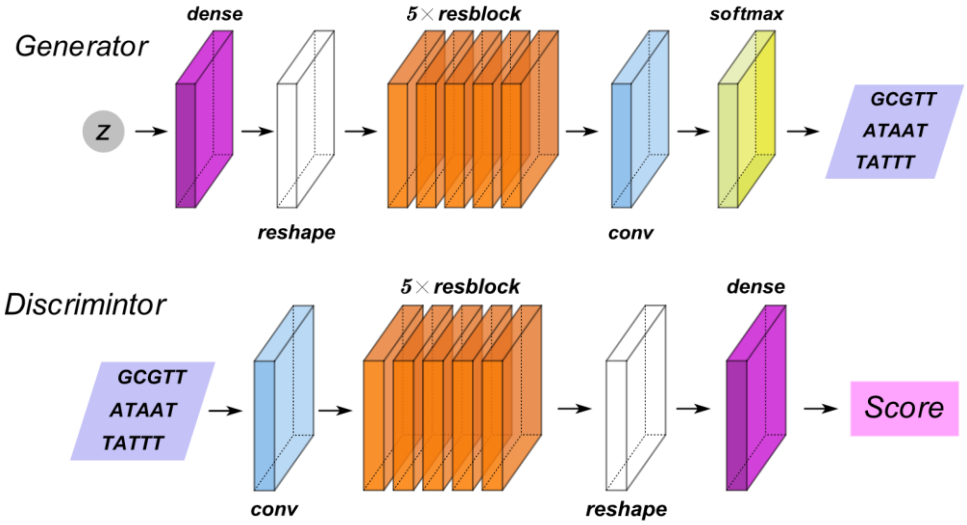
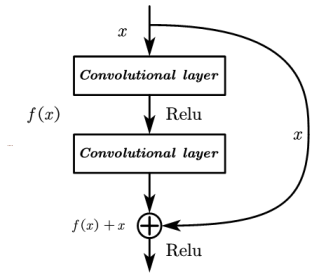


Fig.3 The structure of CNN model for selecting the potential high expression promoter.



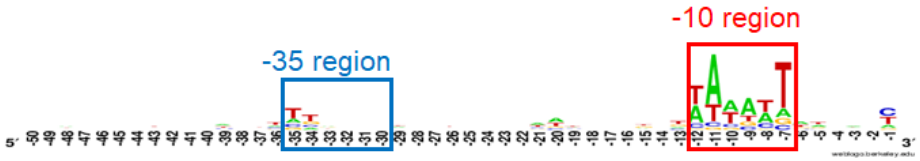
(a)



(b)

Fig.4 (a) The structure of WGAN-GP language model. Different from the DCGAN model, such model introduces resblock and earth moving distance for better generation results. (b) One of the resblock structures are shown here. It is the combination of convolutional layers, and it adds a short connection from resblock start to resblock end.

(a)



(b)

(c)

Fig.5 The sequence logo of (a) natural promoters (b) WGAN-GP synthetic promoters with resblock (c) WGAN-GP synthetic promoters without resblock.The -10 region and -35 region are annotated in the figure.

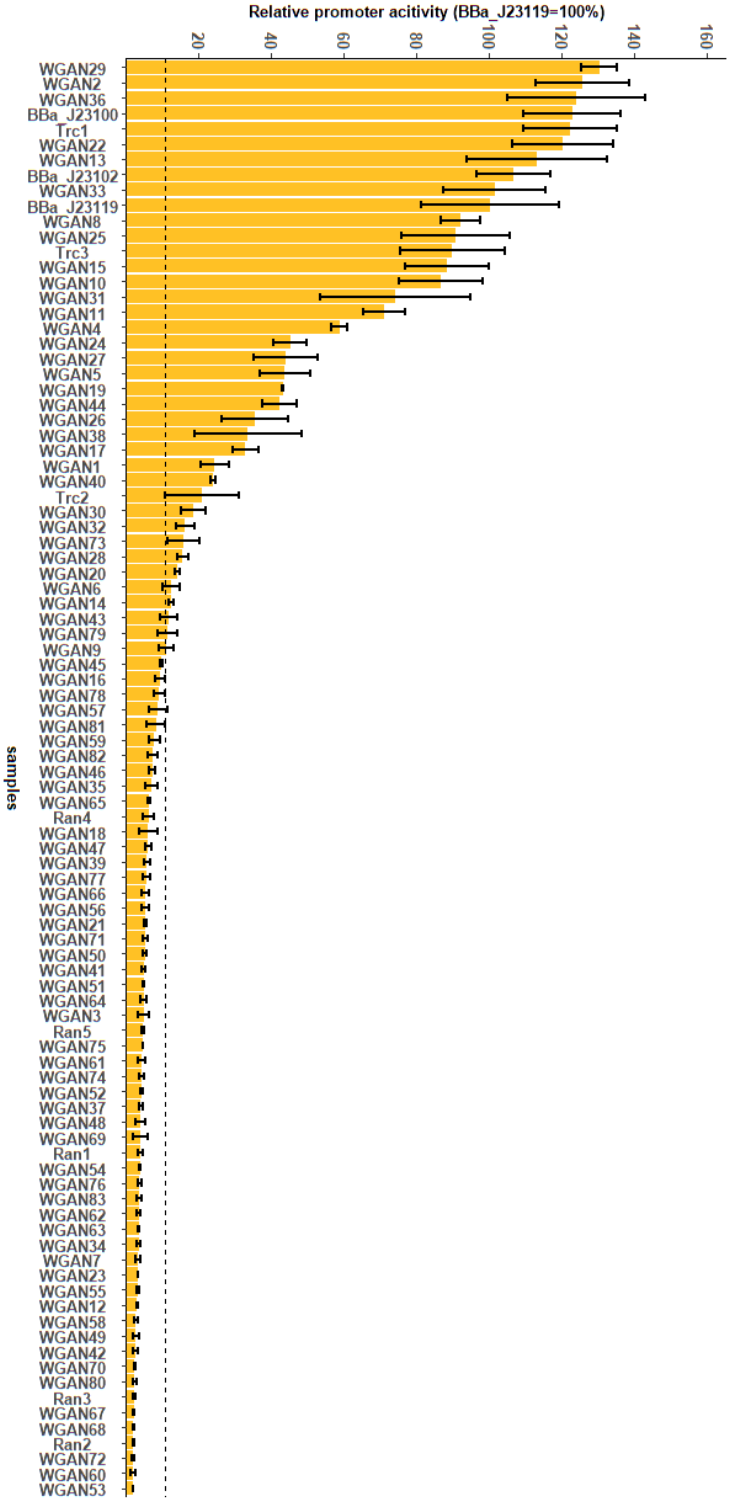


Fig. 6 The 83 new synthetic promoters designed by WGAN-GP exhibit a range of promoter activity. BBa_J23119 and Trc are wild-type promoters , with BBa_J23100、BBa_J23102 and Trc2、Trc3 are their reported promotor mutants showed highest expression in *E.coli* . Rans are five random sequences with equal possibility at each base position with GC content near 50%.

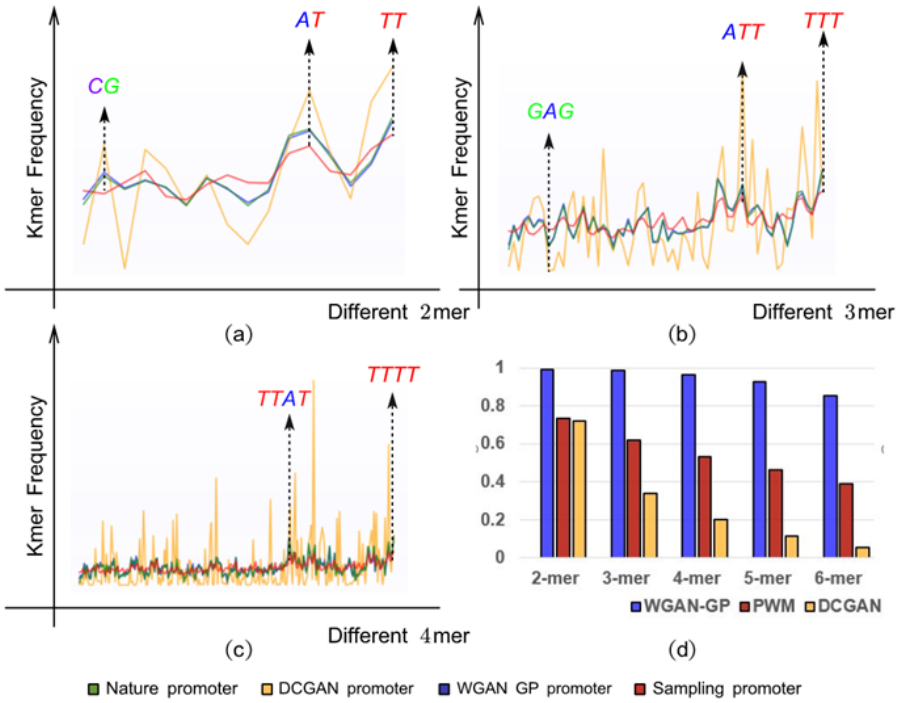
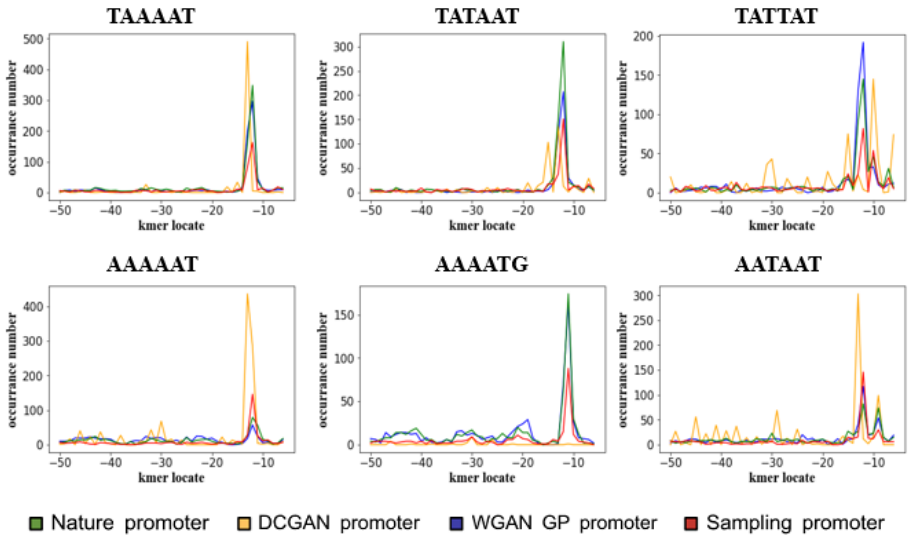
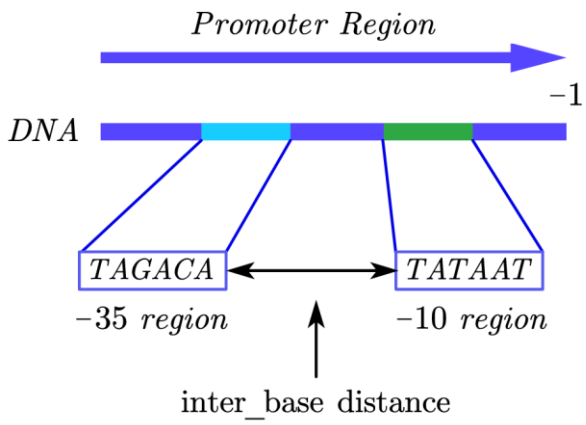


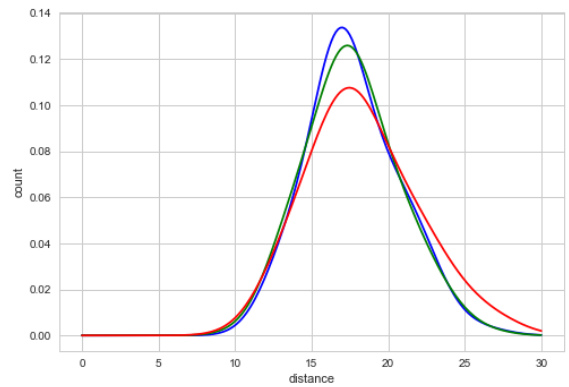
Fig.7 (a)(b)(c) The kmer frequency plots from k = 2 to k = 4, where x-axes represent different kmer. The order of four bases are GCAT, which means if k = 2, then the left 2mer on the x-axis is GG, and the right 2mer is TT. Several high frequency kmers were marked (d) R-squared value evaluates the occurrence number of each kmer between nature promoters and synthetic promoters.



(a)



(b)



(c)

Fig. 8 (a) Five high frequency 6-mer were selected, to illustrate the location learning ability of GAN model. The x-axis represents a certain location, and y-axis represents the number of 6-mer on this location. (b)The definition of inter based distance between –10 and –35 motifs. (c) The inter base distance distribution between the –10 and –35 motifs. The natural promoters are marked in blue and the consensus sequences sampled promoters and WGAN-GP generated promoters are highlighted in red and green respectively.

REFERENCES

1.Lubliner, S., Keren, L., and Segal, E. (2013) Sequence features of yeast and human core promoters that are predictive of maximal promoter activity, *Nucleic acids research 41*, 5569-5581.

2.Mazumder, M., and McMillen, D. R. (2014) Design and characterization of a dual-mode promoter with activation and repression capability for tuning gene expression in yeast, *Nucleic acids research 42*, 9514-9522.

3.Juven-Gershon, T., and Kadonaga, J. T. (2010) Regulation of gene expression via the core promoter and the basal transcriptional machinery, *Developmental biology 339*, 225-229.

4.De Mey, M., Maertens, J., Lequeux, G. J., Soetaert, W. K., and Vandamme, E. J. (2007) Construction and model-based analysis of a promoter library for E. coli: an indispensable tool for metabolic engineering, *BMC biotechnology 7*, 34.

5.Gilman, J., and Love, J. (2016) Synthetic promoter design for new microbial chassis, *Biochemical Society transactions 44*, 731-737.

6.Guiziou, S., Sauveplane, V., Chang, H. J., Clerte, C., Declerck, N., Jules, M., and Bonnet, J. (2016) A part toolbox to tune genetic expression in Bacillus subtilis, *Nucleic acids research 44*, 7495-7508.

7.Nevoigt, E., Kohnke, J., Fischer, C. R., Alper, H., Stahl, U., and Stephanopoulos, G. (2006) Engineering of promoter replacement cassettes for fine-tuning of gene expression in Saccharomyces cerevisiae, *Appl Environ Microbiol 72*, 5266-5273.

8.Du, J., Yuan, Y., Si, T., Lian, J., and Zhao, H. (2012) Customized optimization of metabolic pathways by combinatorial transcriptional engineering, *Nucleic acids research 40*, e142.

9.Portela, R. M., Vogl, T., Kniely, C., Fischer, J. E., Oliveira, R., and Glieder, A. (2017) Synthetic Core Promoters as Universal Parts for Fine-Tuning Expression in Different Yeast Species, *ACS synthetic biology 6*, 471-484.

10.Blazeck, J., Liu, L., Redden, H., and Alper, H. (2011) Tuning gene expression in Yarrowia lipolytica by a hybrid promoter approach, *Appl Environ Microbiol 77*, 7905-7914.

11.Blazeck, J., Garg, R., Reed, B., and Alper, H. S. (2012) Controlling promoter strength and regulation in Saccharomyces cerevisiae using synthetic hybrid promoters, *Biotechnology and bioengineering 109*, 2884-2895.

12.Nishikata, K., Cox, R. S., 3rd, Shimoyama, S., Yoshida, Y., Matsui, M., Makita, Y., and Toyoda, T. (2014) Database construction for PromoterCAD: synthetic promoter design for mammals and plants, *ACS synthetic biology 3*, 192-196.

13.Liu, W., Mazarei, M., Peng, Y., Fethe, M. H., Rudis, M. R., Lin, J., Millwood, R. J., Arelli, P. R., and Stewart, C. N., Jr. (2014) Computational discovery of soybean promoter cis-regulatory elements for the construction of soybean cyst nematode-inducible synthetic promoters, *Plant biotechnology journal 12*, 1015-1026.

14.Alper, H., Fischer, C., Nevoigt, E., and Stephanopoulos, G. (2005) Tuning genetic control through promoter engineering, *Proceedings of the National Academy of Sciences 102*, 12678-12683.

15.Yim, S. S., An, S. J., Kang, M., Lee, J., and Jeong, K. J. (2013) Isolation of fully synthetic promoters for high‐level gene expression in Corynebacterium glutamicum, *Biotechnology and bioengineering 110*, 2959-2969.

16.Davis, J. H., Rubin, A. J., and Sauer, R. T. (2011) Design, construction and characterization of a set of insulated bacterial promoters, *Nucleic acids research 39*, 1131-1141.

17.Sanchez-Lengeling, B., and Aspuru-Guzik, A. (2018) Inverse molecular design using machine learning: Generative models for matter engineering, *Science 361*, 360-365.

18.Wu, K., and Rao, C. V. (2012) Computational methods in synthetic biology: towards computer-aided part design, *Current opinion in chemical biology 16*, 318-322.

19.Budden, D. M., Hurley, D. G., and Crampin, E. J. (2015) Predictive modelling of gene expression from transcriptional regulatory elements, *Briefings in bioinformatics 16*, 616-628.

20.Kiryu, H., Oshima, T., and Asai, K. (2005) Extracting relations between promoter sequences and their strengths from microarray data, *Bioinformatics 21*, 1062-1068.

21.Rhodius, V. A., Mutalik, V. K., and Gross, C. A. (2012) Predicting the strength of UP-elements and full-length E. coli sigmaE promoters, *Nucleic acids research 40*, 2907-2924.

22.Brewster, R. C., Weinert, F. M., Garcia, H. G., Song, D., Rydenfelt, M., and Phillips, R. (2014) The transcription factor titration effect dictates level of gene expression, *Cell 156*, 1312-1323.

23.Brewster, R. C., Jones, D. L., and Phillips, R. (2012) Tuning promoter strength through RNA polymerase binding site design in Escherichia coli, *PLoS computational biology 8*, e1002811.

24.Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016) Improved techniques for training gans, In *Advances in Neural Information Processing Systems*, pp 2234-2242.

25.Seeliger, K., Güçlü, U., Ambrogioni, L., Güçlütürk, Y., and van Gerven, M. (2018) Generative adversarial networks for reconstructing natural images from brain activity, *NeuroImage 181*, 775-785.

26.Gupta, A., and Zou, J. (2018) Feedback GAN (FBGAN) for DNA: a Novel Feedback-Loop Architecture for Optimizing Protein Functions, *arXiv preprint arXiv:1804.01694*.

27.Bonn, S., Machart, P., Marouf, M., Magruder, D. S., Bansal, V., Kilian, C., and Krebs, C. F. (2018) Realistic in silico generation and augmentation of single cell RNA-seq data using Generative Adversarial Neural Networks, *bioRxiv*, 390153.

28.Gama-Castro, S., Salgado, H., Santos-Zavaleta, A., Ledezma-Tejeida, D., Muniz-Rascado, L., Garcia-Sotelo, J. S., Alquicira-Hernandez, K., Martinez-Flores, I., Pannier, L., Castro-Mondragon, J. A., Medina-Rivera, A., Solano-Lira, H., Bonavides-Martinez, C., Perez-Rueda, E., Alquicira-Hernandez, S., Porron-Sotelo, L., Lopez-Fuentes, A., Hernandez-Koutoucheva, A., Del Moral-Chavez, V., Rinaldi, F., and Collado-Vides, J. (2016) RegulonDB version 9.0: high-level integration of gene regulation, coexpression, motif clustering and beyond, *Nucleic acids research 44*, D133-143.

29.Borries, A., Vogel, J., and Sharma, C. M. (2011) Differential RNA Sequencing (dRNA‐Seq): Deep‐Sequencing‐Based Analysis of Primary Transcriptomes, *Tag‐Based Next Generation Sequencing*, 109-121.

30.Thomason, M. K., Bischler, T., Eisenbart, S. K., Forstner, K. U., Zhang, A., Herbig, A., Nieselt, K., Sharma, C. M., and Storz, G. (2015) Global transcriptional start site mapping using differential RNA sequencing reveals novel antisense RNAs in Escherichia coli, *Journal of bacteriology 197*, 18-28.

31.Kelley, D. R., Snoek, J., and Rinn, J. L. (2016) Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks, *Genome research*.

32.Crooks, G. E., Hon, G., Chandonia, J.-M., and Brenner, S. E. (2004) WebLogo: a sequence logo generator, *Genome research 14*, 1188-1190.

33.He, K., Zhang, X., Ren, S., and Sun, J. (2016) Deep residual learning for image recognition, In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 770-778.

34.Arjovsky, M., and Bottou, L. (2017) Towards principled methods for training generative adversarial networks, *arXiv preprint arXiv:1701.04862*.

35.Arjovsky, M., Chintala, S., and Bottou, L. (2017) Wasserstein gan, *arXiv preprint arXiv:1701.07875*.

36.Larkin, M. A., Blackshields, G., Brown, N., Chenna, R., McGettigan, P. A., McWilliam, H., Valentin, F., Wallace, I. M., Wilm, A., and Lopez, R. (2007) Clustal W and Clustal X version 2.0, *Bioinformatics 23*, 2947-2948.

37.Harley, C. B., and Reynolds, R. P. (1987) Analysis of E. coli pormoter sequences, *Nucleic acids research 15*, 2343-2361.

38.Mirza, M., and Osindero, S. (2014) Conditional generative adversarial nets, *arXiv preprint arXiv:1411.1784*.

39.Brock, A., Donahue, J., and Simonyan, K. (2018) Large scale gan training for high fidelity natural image synthesis, *arXiv preprint arXiv:1809.11096*.

40.Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. (2018) Self-attention generative adversarial networks, *arXiv preprint arXiv:1805.08318*.

41.Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks, In *Proceedings of the IEEE International Conference on Computer Vision*, pp 2223-2232.

42.Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A. P., Tejani, A., Totz, J., and Wang, Z. (2017) Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, *computer vision and pattern recognition*, 105-114.

43.Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014) Generative adversarial nets, In *Advances in neural information processing systems*, pp 2672-2680.

44.Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017) Improved training of wasserstein gans, In *Advances in Neural Information Processing Systems*, pp 5767-5777.

45.Killoran, N., Lee, L. J., Delong, A., Duvenaud, D., and Frey, B. J. (2017) Generating and designing DNA with deep generative models, *arXiv preprint arXiv:1712.06148*.