Published online 27 October 2021

https://doi.org/10.1093/nargab/lqab095

Quantifying the tissue-specific regulatory information within enhancer DNA sequences

Philipp Benner* **and Martin Vingron**

Department of Computational Molecular Biology, Max Planck Institute for Molecular Genetics, Ihnestraße 73, 14195 Berlin, Germany

Received May 07, 2021; Revised September 23, 2021; Editorial Decision September 26, 2021; Accepted September 28, 2021

ABSTRACT

Recent efforts to measure epigenetic marks across a wide variety of different cell types and tissues provide insights into the cell type-specific regulatory landscape. We use these data to study whether there exists a correlate of epigenetic signals in the DNA sequence of enhancers and explore with computational methods to what degree such sequence patterns can be used to predict cell type-specific regulatory activity. By constructing classifiers that predict in which tissues enhancers are active, we are able to identify sequence features that might be recognized by the cell in order to regulate gene expression. While classification performances vary greatly between tissues, we show examples where our classifiers correctly predict tissue-specific regulation from sequence alone. We also show that many of the informative patterns indeed harbor transcription factor footprints.

INTRODUCTION

Complex multicellular organisms comprise a large number of different cell types, which all share the same genome. Nevertheless, cell morphology and function are determined by the combination of genes that are expressed (1,2). To unravel how cells control gene expression, we must first identify all regulatory elements of the genome. This is relatively easy for promoters, because they are located proximal to the target gene's transcription start site. Enhancers, which regulate cell type-specific gene transcription, are located distal to transcription start sites and therefore much more difficult to identify. Recent efforts focused on measuring epigenetic marks across a variety of different cell types and tissues (3–5). These marks, including histone modifications, DNA methylation and DNA accessibility, allow one to identify for each cell type regions of the genome that may act as enhancer elements.

However, this information alone can only be regarded as a first step toward understanding the regulatory program of a cell. Ultimately, we would like to understand why certain regions act as enhancer elements in particular cell types and how they drive gene expression. We believe that this information must be encoded in the genome in the form of binding sites for proteins such as transcription or pioneer factors. Hence, it should be possible to use the genomic DNA sequence not only for identifying enhancer elements but also for predicting the cell types in which the elements are active. Unfortunately, our knowledge of transcription factors, including their DNA binding preferences and interactions, remains incomplete. Instead, our goal is to quantify how much information about the cell type-specific activity of an enhancer element is actually encoded in the DNA sequence. For this, we may use enhancers identified from epigenetic marks and train classifiers that predict for each element the cell type in which it was found to be active. The classification performance then allows us to quantify to what extent there exists a correlate of cell type-specific epigenetic marks in the DNA sequence of enhancer elements.

An increasing number of studies focus on the prediction of regulatory elements from DNA sequence. Especially, the DNA sequences of promoters have been studied in depth and several cell type-specific binding patterns have been identified (6–10). Since regulatory regions can be better targeted by transcription factors when they are not concealed by nucleosomes (11), other studies focused on the prediction of accessible regions as measured by DNase-seq (12) or ATAC-seq (13). Within cell types, accessible regions can be predicted from DNA sequence alone with high accuracy (14). Especially for ubiquitous regions, which are open in many or all cell types, the accuracy is very high. An analysis of feature importance revealed motifs of pioneer factors as well as CpG dinucleotide content (14). Other studies focused on the genome-wide prediction of active enhancers from DNA sequence (15,16), which were identified through either enhancer-specific patterns of histone marks or ChIP-seq experiments targeting EP300. The overall performance of such methods is good and comparable to the performance of open chromatin predictions. The focus of these studies, however, lies on the genome-wide prediction of enhancer elements (14–17) within a particular cell type. Methods developed for this task are trained to separate regula-

*To whom correspondence should be addressed. Tel: +49 30 8413 1150; Fax: +49 30 8413 1152; Email: benner@molgen.mpg.de tory elements from other genomic sequences and may identify sequence patterns common to all regulatory elements or the genomic background.

Instead, we pursue a different line of research that focuses on learning tissue or cell type-specific patterns (17,18). Our learning setup consists of enhancer DNA sequences active in one or more tissues (positive set) versus enhancer DNA sequences from all remaining tissues (negative set). We use the recently developed leapfrog logistic regression (19) on sequence k-mers that allows to efficiently compute sparse solutions on very high dimensional data. From the trained classifiers, we quantify how much cell type-specific regulatory information is actually contained in the DNA sequence of enhancers. Furthermore, the classifiers are easy to interpret and allow to identify DNA subsequences or code words that may drive cell type-specific gene expression. We use ATAC-seq data to validate our findings by showing that many of the informative patterns indeed harbor transcription factor footprints (Figure 1).

Identified code words are only meaningful if the training set is of high quality and reflects our research objectives. Our interest lies in tissues as opposed to isolated cell lines that sometimes are derived from immortalized cancer cells. ENCODE provides comparable epigenetic data across several tissues in the mouse embryo. We use ModHMM (20) for computing genome segmentations based on ATAC-seq, RNA-seq and ChIP-seq data for multiple histone modifications. ModHMM computes highly accurate annotations of active enhancers that are not tainted by promoter elements or inactive (e.g. primed) enhancers.

MATERIALS AND METHODS

Enhancer identification with ModHMM

ModHMM (20) is a hidden Markov model for computing genome segmentations similar to ChromHMM (21) and EpiCSeg (22). We use ModHMM because a reliable discrimination between enhancers and promoters is vital to this study. A comparison of classification results on promoters can be found in the Supplementary Data. It uses a fixed number of hidden states and computes segmentations based on eight features, namely ATAC/DNase-seq and RNA-seq in addition to histone modifications H3K4me1, H3K4me3, H3K27ac, H3K27me3 and H3K9me3. Compared to other genome segmentation methods, ModHMM better discriminates between different types of regulatory elements, such as active promoters and enhancers, by incorporating prior knowledge into the model. In a nutshell, there is only one state associated with active enhancers and the epigenetic signals known to mark active enhancers are encoded in the model. In contrast to ChromHMM, ModHMM also models the spatial distribution of epigenetic marks around regulatory elements. Furthermore, it implements a constrained Markov model that respects the grammar of the genome; i.e. the active promoters must be flanked by a transcribed region.

ModHMM version 1.2.3 was used to compute genome segmentations of eight embryonic mouse tissues with a bin size of 200 bp. All segmentations are available online at https://github.com/pbenner/modhmm-segmentations, from which we obtained the tissue-specific positions of active enhancers. For each type of regulatory element, the eight lists of tissue-specific genomic positions were merged by joining all elements that overlap by at least one bin. The size of all elements was then set to 1000 bp around the center and we acquired the DNA sequences of all regulatory elements from the mm10 assembly.

All differential regulatory elements are available for download at https://doi.org/10.5281/zenodo.5112066.

Logistic regression classifiers

We use leapfrog regularization (19) for estimating logistic regression classifiers on k-mers (KLR). The parameters $\theta \in \mathbb{R}^{m+1}$ of the classifier are estimated by maximizing the ℓ_1-penalized log-likelihood function

$$\log l_\lambda(\theta) = \sum_{i=1}^{n} \{y_i w_1 \log \sigma(x_i\theta) \\ +(1-y_i)w_0 \log(1-\sigma(x_i\theta))\} + \lambda||\theta||_1,$$

with class weights

$$w_0 = \frac{n}{2\sum_{i=1}^{n}(1-y_i)}, \quad w_1 = \frac{n}{2\sum_{i=1}^{n} y_i},$$

on a set of n observations $x = \{x_1, x_2, \ldots, x_n\}$ with labels $y_i \in \{0, 1\}$. Each $x_i = (1, x_{i1}, x_{i2}, \ldots, x_{im})$ is a row vector of length $m + 1$, where m is the number of features. In the above formula, σ denotes the sigmoid function. $\lambda \in \mathbb{R}_{\geq 0}$ is a parameter that controls the strength of the penalty. In practice, it is difficult to select appropriate values for the penalty λ. A common practice is to simply test a fixed set of values for λ. We avoid this by using leapfrog regularization (19), which determines λ so that a predefined number of features q are selected, i.e. $\|\theta\|_0 = q + 1$ for $q \leq m$. Please note that although we wrote $\|\theta\|_1$ in the above formula, we actually do not regularize the first component of θ, i.e. $||\theta||_1 = \sum_{j=2}^{m+1} |\theta_j|$. To compute maximum likelihood solutions, we implemented a just-in-time variant (23) of the SAGA algorithm (24).

Although we have multiple classes, i.e. tissues, we rely on simple binary classification and compare classification results for multiple splits of the data set into positive and negative classes. Multiclass logistic regression is feasible only if class probabilities are constrained, i.e. if they are required to sum up to 1. However, this constraint is not applicable here since enhancers are often active in multiple tissues.

For the KLR classifier, the set of features consists of all k-mers of length 4–8 with any number of gaps (denoted by N). Reverse complements are considered equivalent; i.e. the vector x_i has a single coordinate for both ANNTG and its reverse complement CANNT, denoted by ANNTG—CANNT. Hence, each x_i is a vector of dimension 156 570 + 1 ($m = 156\,570$). We consider either code word counts or occurrences (binarized counts). In the case of code word counts, we first normalize the data to unit variance. The data are not centered at zero to retain the sparse structure.

For the entire study, classifier performance is evaluated using 10-fold cross-validation. We do not know the optimal number of features (nonzero coefficients). Therefore, during each CV iteration, 10% of the training data are reserved as a

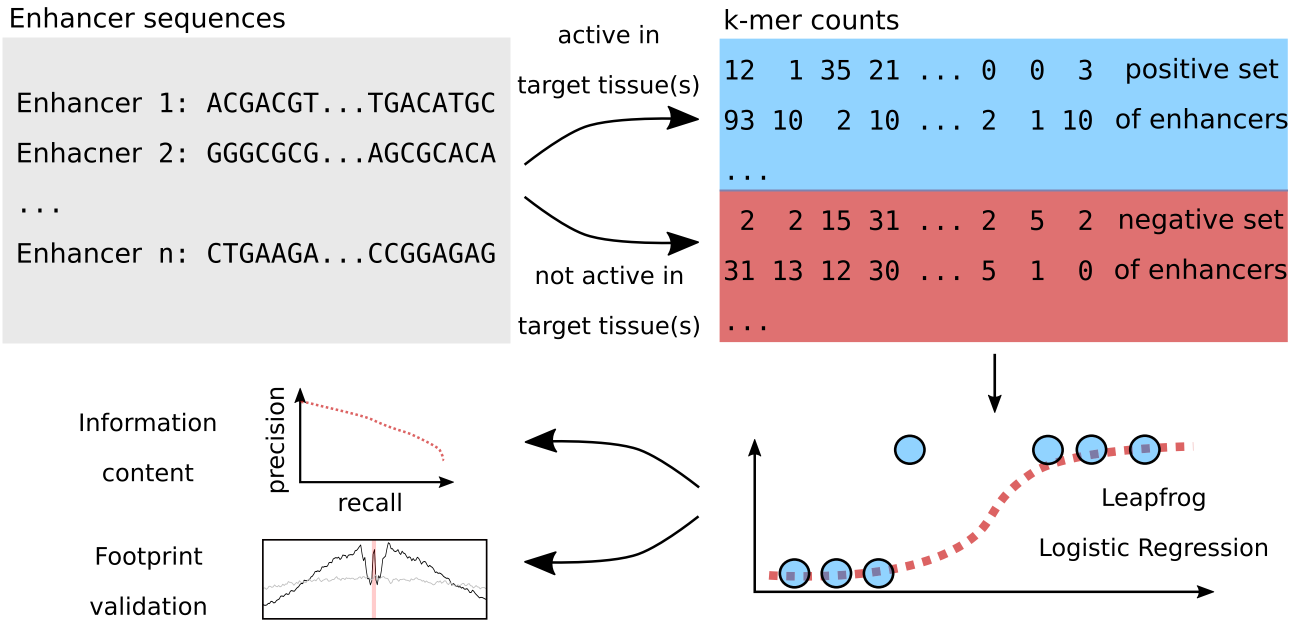

Figure 1. Overview. We use enhancer sequences from multiple tissues and extract k-mer occurrences. Logistic regression classifiers are trained with leapfrog regularization. From the performance of trained classifiers, we quantify cell type-specific regulatory information within DNA sequences. Furthermore, we extract cell type-specific patterns and validate them through ATAC-seq footprints.

validation set for selecting the best q. For the KLR classifier, we tested $q = 10, 100, 200, \ldots, 900, 1000, 2000, \ldots, 6000$.

Our software is available at https://github.com/pbenner/kmerLr.

k-mer SVM

LS-GKM (25) is a support vector machine (SVM) that also uses gapped k-mer frequencies as features. We mainly use it to benchmark the performance of the KLR classifier. However, the decisions that SVMs make are difficult to interpret and extracting important features from it is much more complicated than for the KLR classifier.

Clustering of tissues

The clustering of tissues is computed from ModHMM segmentations, where only the enhancer state is used. First, a similarity matrix between the tissues is computed by counting the number of overlapping enhancers between each pair of tissues. Afterward, this matrix is used to compute a hierarchical clustering (based on the *hclust* method in R). The clustered data collection is computed from this clustering in the following way. By removing an inner edge from the hierarchical clustering tree, a bipartition of tissues is generated. This bipartition corresponds to a single data set of the collection and is used for labeling the enhancers as positive or negative. Enhancers that appear in both the positive and negative sets are removed.

RESULTS

Characteristics of data sets

We obtained data from eight tissues (heart, kidney, liver, limb, lung, forebrain, midbrain and hindbrain) of embryonic mouse at day 15.5 from the ENCODE project (Supplementary Tables S27 and S28). ModHMM (20) was used to compute genome segmentations for each of the tissues. The segmentations provide, among others, the genomic coordinates of several types of regulatory elements within each tissue, such as active promoters and enhancers as well as primed regions. Across tissues, regulatory elements that overlap are merged and we record the set of tissues in which the element was observed. To avoid biases, we set the length of all regulatory regions to 1000 bp around the center. Figure 2 shows the tissue specificity of promoters and enhancers. Most promoters are active in all eight tissues, which supports similar findings from RNA-seq studies (26). We also find that the observed-to-expected CpG ratio (27), referred to as CpG ratio in the following, grows steadily with the number of tissues in which promoters are active. On the other hand, most enhancers are active only in very few tissues and there is almost no enhancer active in all eight tissues. These results suggest that for genes that are active in multiple tissues there exists a distinct set of enhancers in each tissue that drives expression. Many enhancers and promoters in forebrain, midbrain and hindbrain are active in two other tissues (i.e. Figure 2 shows a peak for brain tissues at three), reflecting the fact that brain tissues are very similar and share many regulatory elements. The tissue specificity of primed enhancers is very similar to active enhancers (Supplementary Figure S3).

Learning setup

As shown in the previous section, most enhancers are active in only a few tissues, which indicates that enhancers drive cell type-specific gene expression. A more detailed analysis on the location of the cell type-specific regulatory code can be found in Supplementary Section S1.3. Hence, we focus in the following on the analysis of enhancer DNA sequences.

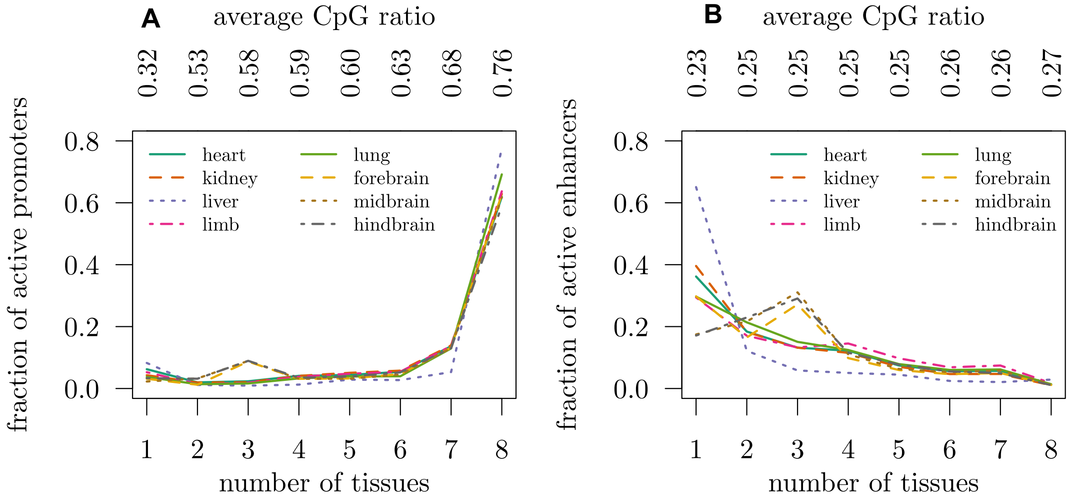

Figure 2. Tissue specificity of regulatory regions. The figures show the fraction of promoters (**A**) and enhancers (**B**) that are active in $n \in \{1, 2, \ldots, 8\}$ tissues, stratified by tissue. For instance, 65% of liver enhancers are active only in liver, but almost no enhancer is active in all eight tissues. In hindbrain, around 30% of the enhancers are active in two other tissues. Above the figure, the average observed-to-expected CpG ratio is shown.

Table 1. Definition of clustered data collection

Data set	Positive	Negative
1	Liver	Heart, kidney, limb, lung, forebrain, midbrain, hindbrain
2	Kidney, lung	Heart, limb, liver, forebrain, midbrain, hindbrain
3	Heart, kidney, lung	Limb, liver, forebrain, midbrain, hindbrain
4	Forebrain, midbrain, hindbrain	Heart, kidney, limb, liver, lung
5	Midbrain, hindbrain	Heart, kidney, limb, liver, lung, forebrain

Enhancer regions were clustered using a hierarchical clustering method. An edge of the resulting tree defines a bipartition of the tissues. Each data set in this collection corresponds to an inner edge of the tree.

For testing different feature sets, we constructed an unbalanced data collection using a one-versus-rest strategy. It consists of eight data sets, each of which defines the enhancer elements of a single tissue as positive samples and the elements of all remaining tissues as negative. We refer to this collection as the *leaves data collection*. Furthermore, we constructed a second data collection where positive and negative sets consist of several tissues grouped together. The *clustered data collection* is defined in Table 1 and visualized as a tree in Figure 4A. For each data set in this collection, tissues were either assigned to the positive or negative set, based on a hierarchical clustering of enhancer regions (see the 'Materials and Methods' section). An enhancer is considered a positive sample if it is active in any of the positive and none of the negative tissues. For both the leaves and clustered data collection, our motivation is to find features specific to particular tissues. For instance, if we want to find forebrain-specific sequence patterns in active enhancers, we construct a classifier that can discriminate between forebrain enhancer sequences (positive set) and enhancer sequences from all remaining tissues (negative set). Especially, the choice of the negative set is very important. To justify why we consider all remaining tissues as negative set, consider the following two cases. First, we only use other brain tissues as negative set. In this case, any identified pattern might be specific to forebrain, but we cannot be certain that it does not occur in other tissues, which is why we must include other tissues in the negative set. Second, we use other non-brain regions as negative set, but then we might identify patterns that are specific to all brain regions instead of just forebrain. Therefore, it is essential to include as many tissues in the negative set as possible. In addition, we must exclude any enhancers that are active in both forebrain and any tissue in the negative set, because these regions might be activated by sequence features that we are not interested in.

In this study, we mainly rely on logistic regression as classifier that we train with the recently developed leapfrog regularization (see the 'Materials and Methods' section). We tested motif scores and *k*-mers as features, but found that *k*-mers generally yield better results (see Supplementary Section S1.1). For logistic regression with *k*-mers (KLR), we consider all *k*-mers of length 4–8 with any number of gaps (denoted by N), which we also call *gapped k-mers*. A *k*-mer and its reverse complement are considered equivalent; i.e. the feature vectors have a single coordinate for both, say, ANNTG and its reverse complement CANNT. We denote this pair as ANNTG—CANNT and refer to it as *code word*. The KLR classifier uses mainly code word counts as features. However, for extracting most important code words from our classifiers, we only use code word occurrences as features, i.e. 1 if a code word appears in an enhancer sequence and 0 otherwise. This simplifies the interpretation of classifiers without much reducing their performance. To gauge the performance of the KLR classifier, we use an SVM with gapped *k*-mer string kernel (25).

Classifier choice and comparison

We first tested the performance of the KLR and SVM classifiers on active enhancers from the clustered collection. We tested the classifiers on each data set of the collection using 10-fold cross-validation (Supplementary Figure S4). Both classifiers perform relatively well with a median area under precision–recall curve (PR-AUC) of 0.71 and 0.67, respectively.

The performances of the KLR and SVM classifiers are similar and both could be used in this study. However, the KLR classifier is slightly better and much easier to interpret; i.e. the parameters of the KLR classifier can be directly linked to the importance of individual *k*-mers. The interpretation of SVMs, on the other hand, is much harder since there exists no natural way of extracting the importance of *k*-mers (28). In addition, training SVMs is computationally much more expensive. We therefore focus mainly on the KLR classifier in the following.

We then tested the KLR classifier on a positive set of active enhancers and a negative set of equal size consisting of random genomic regions in order to evaluate the accuracy of genome-wide predictions. The random regions have the same length as the enhancers, i.e. 1000 bp. We did not exclude any genomic regions, such as repetitive elements, for constructing the negative set. In particular, this ensures that our results are comparable to related studies (14) that present methods for genome-wide predictions of functional elements. The performance is overall very good across the eight different tissues with ROC-AUC values ranging from 0.90 to 0.97 and a median of 0.96 (Figure 3). This result is consistent with previous studies aiming at recognition of open chromatin regions (14–17). Our results confirm that genome-wide predictions of functional elements are relatively easy. Furthermore, we observed that, except for heart, important code words are highly AT-rich (Supplementary Tables S2–S9).

To check the performance results, we tested our cross-validation scheme on a control data set that contains enhancers from all tissues both as foreground and as background. The positive class consists of all enhancers on chromosomes 3, 5, 7, 11, 13, 17 and 19; the enhancers from the remaining chromosomes form the negative set. As expected, the classification performance on this data set is very close to random (Supplementary Figure S5).

Quantification of tissue-specific information

Our main interest is the quantification of tissue-specific information within enhancers. The construction of data sets is essential for extracting this information. If, for instance, we use enhancers active in a particular tissue as positive set and random genomic regions as negative set (as mentioned earlier), then it suffices that classifiers learn patterns common to all enhancers or the genomic background, because it is highly unlikely that the negative set contains enhancers that are active in other tissues. Most related studies (14–17) use random genomic regions as negative set. Instead, we train classifiers on data sets that contain DNA sequences of enhancers active in a particular subset of tissues (positive set) and inactive in all other tissues (negative set). This choice of the negative set is essential for truly learning tissue-specific information. Furthermore, we drop all enhancers that are active in both the positive and negative sets, as those per se do not contain any information specific to tissues in the positive or negative set.

The classification performances can be used as a proxy for the tissue-specific information contained in the DNA sequences of enhancers. This proxy provides only a lower bound on the tissue-specific information, as there might exist classifiers with a higher predictive accuracy. For quantifying this information, we use the KLR classifier, a simple logistic regression on *k*-mers, because it performs as well as SVMs and at the same time is easy to interpret. We consider two different scenarios: the positive set consists of either multiple tissues (clustered data collection) or only a single tissue (leaves data collection).

The classification performance of the KLR classifier on the clustered data collection of active enhancers is shown in Figure 4. We use precision–recall curves instead of ROC curves, because the data sets are unbalanced. In general, the performances are much lower than those for discriminating enhancers from random genomic regions. Separating brain from non-brain enhancers seems to work best. Much harder is the task of separating hindbrain and midbrain enhancers from other tissues, mainly because hindbrain and midbrain enhancers seem to share many sequence features with forebrain enhancers. We also tested cross-validation with test sets sorted by chromosomal position (29) to see whether there is a possible bias in our analysis and found only minor deviations in classification performance (Supplementary Figure S6).

Even more difficult is the discrimination of active enhancers from a single tissue versus all other tissues. Figure 5 shows the performance of the KLR classifier on the leaves data collection. The decline in performance might be caused by the much larger negative sets. Furthermore, the negative set contains enhancers from tissues that are very similar to the one of the positive set. The prediction works best for forebrain and liver. The classifier trained on hindbrain shows the worst performance. This is surprising, because forebrain and hindbrain are both brain tissues, yet the classification performances are quite different. Nevertheless, we show in the following that all classifiers successfully identified important sequence patterns. We also tested the performance of active versus primed enhancers and observed similar prediction accuracies (Supplementary Figure S10).

Code word counts versus occurrences

The KLR classifier can use as features either code word counts or occurrences. The former gives the number of times a code word is observed in the DNA sequence, while the latter is 1 if the code word is present and 0 if it is not. We wanted to understand whether there is additional information in the code word counts, i.e. whether transcription factors recognize single binding sites or whether the overall sequence affinity (30,31) is of importance. On the leaves data collection, we found that the KLR classifier did not perform significantly worse if we only consider code word occurrences instead of counts. We observe an average decline in performance of about 0.02 across tissues (Supplementary

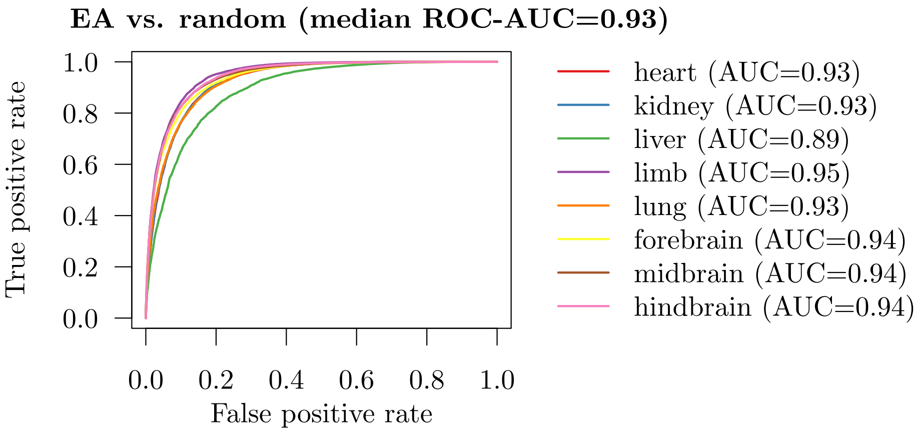

Figure 3. KLR classifier performance on active enhancers (EA) versus random regions. ROC curves were computed using 10-fold cross-validation.

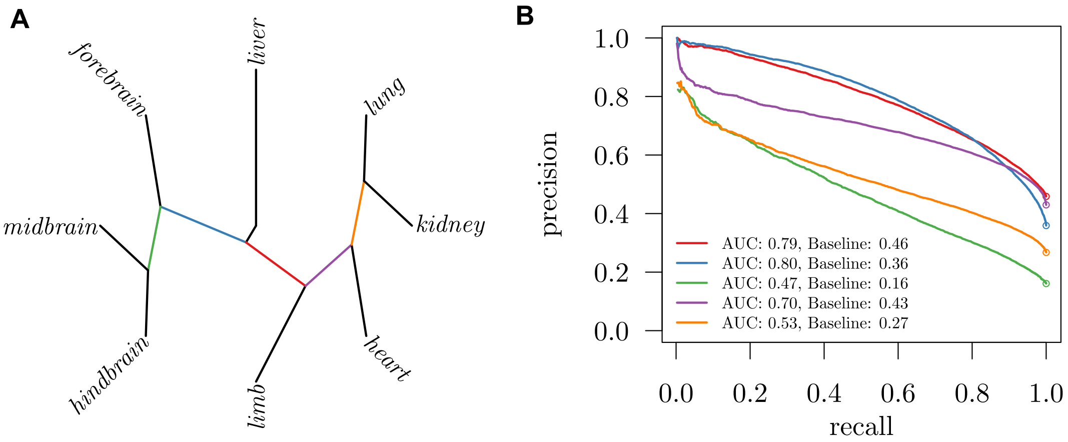

Figure 4. KLR classification performance on active enhancers from the clustered data collection. Panel (**A**) shows a clustering of the eight tissues. Every edge of the tree bipartitions the data into positive and negative samples. Respective classification performances using 10-fold cross-validation are shown in panel (**B**). Edges and corresponding precision–recall curves share the same color. Baseline performances are visualized as circles.

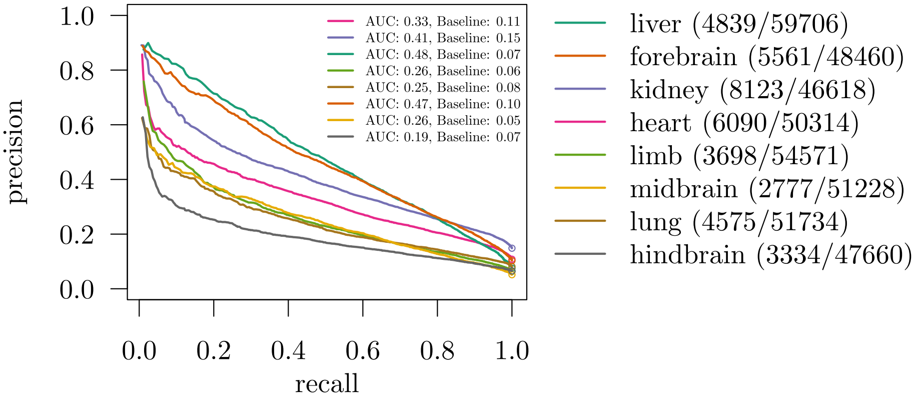

Figure 5. KLR classification performance on active enhancers from the leaves data collection. Ten-fold cross-validation is used to evaluate precision–recall curves. Baseline performances are visualized as circles. The number of enhancers in the positive and negative (pos./neg.) set is reported in the legend.

Table 2. Tissue-specific code words

Tissue	First code word		Second code word
Heart	CTTATC—GATAAG	(GATA)	AGGTCA—TGACCT
Kidney	CNNTGACC—GGTCANNG	(RXR-α/VDR)	ATTAAC—GTTAAT
Limb	CTTGGC—GCCAAG	(NFATc2)	ANTTTCCA—TGGAAANT
Liver	CTTATC—GATAAG	(GATA)	AGATAA—TTATCT
Lung	TAAACA—TGTTTA	(Fox)	GTAAAC—GTTTAC
Forebrain	CAGATGG—CCATCTG	(NeuroD1)	CAATTA—TAATTG
Midbrain	TATTCA—TGAATA	(SOX)	AATAAT—ATTATT
Hindbrain	TAATNAA—TTNATTA		ATNNATCA—TGATNNAT

Two most important tissue-specific code words in active enhancers extracted from KLR classifiers with 100 nonzero coefficients ($q = 100$). Corresponding names of transcription factor matches in the Jaspar database are given in parentheses if a unique assignment was possible.

Figure S13). However, it is possible that the loss of count information is counteracted by the use of more code words. Indeed, we observe an increase in the number of features used by the optimal classifiers for enhancers (Supplementary Figure S9). For our purposes, using only code word occurrences is highly desirable, because it simplifies the interpretation of estimated classifiers, as discussed in the following section.

Identification of cell type-specific regulatory code words in enhancers

To identify code words that might be recognized by transcription factors to drive cell type-specific gene expression, we use the KLR classifier applied to the leaves data set. The coefficients of the logistic regression can be understood as the importance of the corresponding code words. However, even when data are standardized, the interpretation of coefficients is much simpler when only code word occurrences are used (32). Therefore, we extract features from KLR classifiers that use only code word occurrences. Binarizing the data does not seem to lead to a significant drop in performance.

Features are extracted from classifiers trained on the leaves data collection, because those contain information about single tissues. We generally use 10-fold cross-validation to evaluate predictive performances. Features are extracted by first merging the 10 classifiers estimated during cross-validation. For each feature, we take the coefficient with the minimum absolute value across all classifiers, which has the effect that only those features remain that have a nonzero coefficient across all training sets. Code words with high absolute coefficients are typically stable across all cross-validation iterations (Supplementary Figure S12). We report only code words with positive coefficients, because those correspond to k-mers relevant to a single tissue, i.e. the positive set. The results must be interpreted with caution, because single code words do not determine cell type-specific activity. We only observe that weighted sets of code words are predictive of tissue-specific enhancer activity with varying levels of accuracy.

Table 2 shows the first two most important code words; a complete list is provided in Supplementary Tables S10–S17. We used Tomtom (33) to search for matching motifs in the Jaspar database (34). For kidney, we identified a code word that may belong to PPAR-γ, ER1 or the RXR-α/VDR heterodimer; the latter has known functions in kidney (35). The limb classifier identified a code word that probably belongs to NFI-C binding sites, which is highly expressed in skeletal muscle cells (36). For heart and liver, the most important code words seem to belong to the GATA family of transcription factors, while for lung we identified a code word that is most likely recognized by members of the Fox family. The top forebrain code word is a known binding site for NeuroD1, a neurogenic differentiation factor (37). We also identified code words that are not associated with any transcription factor. Some contain two or more gaps (e.g. TAATNAA—TTNATTA or ATNNATCA—TGATNNAT), which might reflect co-binding sites of transcription factors. A comprehensive list of code word matches with transcription factor motifs is given in Supplementary Table S26.

As a further control, we looked at ATAC-seq footprints (38) around code word occurrences in active enhancers. We first computed the ATAC-seq coverages by treating paired-end reads as single end and reducing read lengths to 2 bp. Forward strand reads were shifted by 4 bp and reverse strand reads by −5 bp. For each tissue, we scanned all active enhancer sequences for occurrences of the most important code word. Enhancers that do not contain the most important code word were omitted. We then aligned the ATAC-seq signal around the code word positions. When an enhancer contained the code word more than once, the position of the first occurrence was used. Except for forebrain, we observe clear ATAC-seq footprints at the centers (Figure 6), which suggests that the identified code words are indeed recognized by transcription factors. Most of the top-scoring code words show similar footprints (Supplementary Figures S14–S21). Some footprints show a single valley, while others show a peak at the center, which is surrounded by two valleys. The latter signal might stem from cooperative binding of transcription factors. In addition, Figure 6 shows control footprints from code word occurrences in the mitochondrial genome (Supplementary Section S1.5). For the most important code word in hindbrain, we observe a similar footprint in the mitochondrial genome. This result suggests that the observed footprint might not be caused by transcription factor binding. Furthermore, we used the same aligned sequences to compute logos of code word neighborhoods and observe almost no signal outside the code words (Supplementary Figure S11). This result suggests that a clustering of code words might not be possible.

Sliding window predictions at selected loci

So far, we have extensively quantified the predictive accuracy of our classifiers. In a nutshell, genome-wide predictions of active enhancers are relatively easy (i.e. active enhancers versus random genomic regions). More difficult is

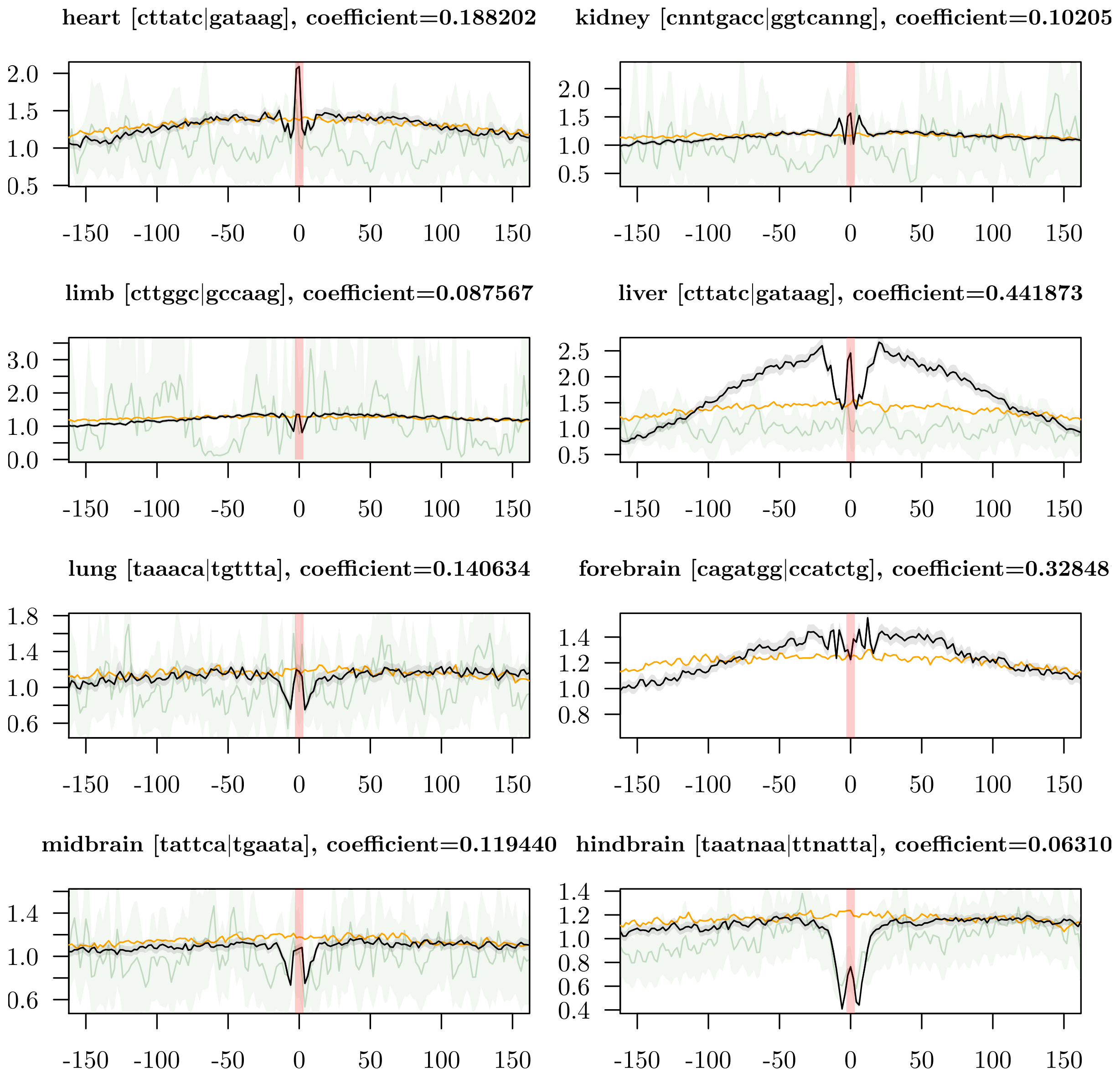

Figure 6. ATAC-seq footprints around most important code words of active enhancers (black lines with confidence interval of two standard deviations as shaded area). Regions are aligned around the center of code word occurrences and the footprint is computed as the average over all *n* active enhancers in the given tissue. The distance in bp from the center is shown on the *x*-axis. Red shaded areas show the positions of the code words. The control (orange lines) shows the average ATAC-seq signal of the same regions, but aligned at the center of the enhancer elements. ATAC-seq footprints around code word occurrences in the mitochondrial DNA are shown as a second control (green line with confidence interval of two standard deviations as shaded area). This control is not shown if the mitochondrial DNA does not harbor the corresponding code word.

the prediction of tissues in which an enhancer is active. Here, we demonstrate our findings on a single loci, where we compute sliding window predictions of active enhancers. More specifically, a sliding window of 2 kb is used to compute predictions of brain and liver enhancers along the genome. We first use a classifier that discriminates between enhancer elements and random genomic regions and combine this with a second classifier that separates brain from liver enhancers. Figure 7 shows a region between Cdh9 and Cdh10, which in humans is known to harbor several enhancer elements and where genetic variants are associated with autism spectrum disorders (39,40). We excluded this locus from the training sets of the classifiers. The genome segmentation identified two enhancers that are active only in forebrain.

The sliding window prediction of brain enhancers shows a peak at only one of the active enhancers, but also very few false positives within this locus. This observation is in line with our previous results. We showed that random genomic regions can be easily distinguished from enhancers, but differentiating between active enhancers in particular cell types is hard. However, the sliding window predictions

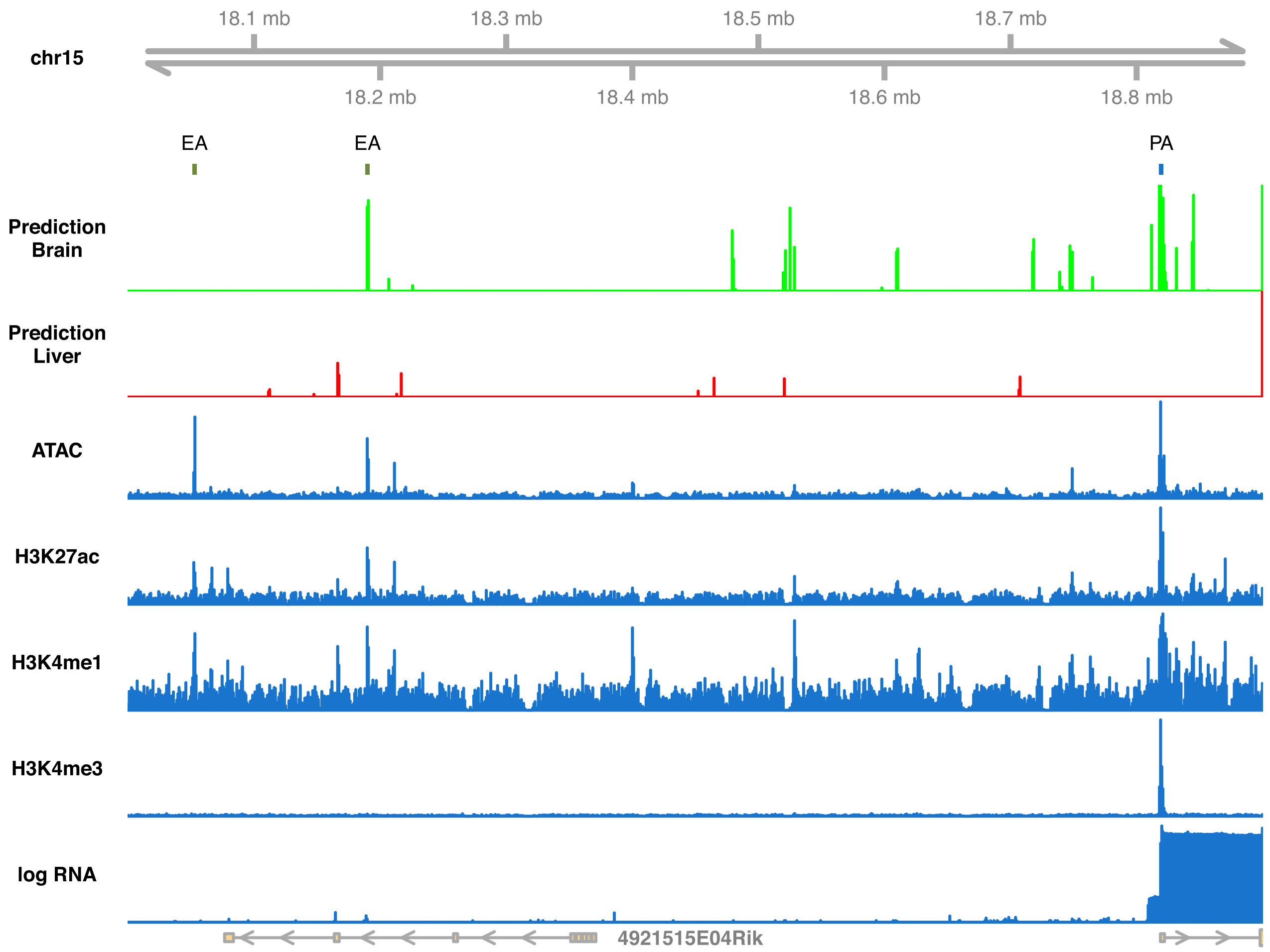

Figure 7. Sliding window predictions of active enhancers from DNA sequence. ModHMM-predicted active promoters (PA) and enhancers (EA) in forebrain are marked by small bars in blue and green, respectively. Predictions of active brain enhancers from DNA sequence are shown in green, and predictions of liver enhancers in red. The blue tracks below show ATAC-seq, histone marks and RNA-seq coverages in forebrain.

also detect the active promoter of Cdh10, possibly because we did not train the classifier to discriminate between promoters and enhancers. Nevertheless, the precision of predictions is very promising and rarely reported in the literature. By inverting the predictions of the second classifier (probability of complement), we obtain enhancer predictions for liver. The prediction of liver enhancers shows no peaks at the brain enhancers or the Cdh10 promoter. Throughout this locus, the probability of liver enhancers is very low.

DISCUSSION

In this study, we constructed classifiers that predict tissue-specific enhancer activity from DNA sequence. This has been done by several other studies (14–17), which focus on the genome-wide prediction of enhancers. We show that this task is relatively easy, because such classifiers mainly learn to discriminate between enhancer elements and other genomic regions. Instead, we focus on predicting cell type-specific activity of enhancer elements, which we achieve by training classifiers that discriminate between enhancers active in selected tissues. We use the classification performance as a proxy to measure how much information about cell type-specific activity is contained in the DNA sequence of enhancers. By using classifiers that are easy to interpret, we were able to extract important regulatory code words that might be recognized by transcription factors for driving cell type-specific gene expression. The ATAC-seq signature around identified code words shows a clear pattern, which indicates that we indeed identified functional binding sites. Furthermore, the accuracy of our sliding window predictions is very promising and rarely reported in the literature.

The classification performance of active enhancers strongly depends on the tissues we want to discriminate. Our classifier performs well when discriminating between highly dissimilar tissues, such as brain and non-brain tissues. However, especially when the positive class consists of enhancers from a single tissue, the performance drops in many cases. One possible reason is that the tissues we are dealing with contain too many different cell types or the data are simply too noisy. Another explanation is that we still have only a poor understanding of the cell type-specific regulatory code and the features required for predicting the activity of enhancer elements. It could be that not all the required information is contained in the DNA sequence of isolated enhancers and we are still missing a piece of the puzzle.

There are many possible future research directions. For instance, single-cell ATAC-seq data could help to distinguish between cell types within a tissue and thereby help to train better classifiers. Furthermore, we know that transcription factors and other proteins bound to enhancers and promoters interact in order to initiate transcription. It is possible that the regulatory code is distributed among enhancers and promoters and that both must be considered jointly when training classifiers.

SUPPLEMENTARY DATA

Supplementary Data are available at NARGAB Online.

ACKNOWLEDGEMENTS

We thank Stefan Haas and Kirsten Kelleher for their comments on the manuscript and many inspiring discussions.

FUNDING

German Ministry of Education and Research (BMBF) [01IS18037G].

Conflict of interest statement. None declared.

REFERENCES

1. Alberts,B. (2017) *Molecular Biology of the Cell*. CRC Press, Boca Raton, FL.
2. Ralston,A. and Shaw,K. (2008) Gene expression regulates cell differentiation. *Nat. Educ.*, **1**, 127–131.
3. Stamatoyannopoulos,J.A., Snyder,M., Hardison,R., Ren,B., Gingeras,T., Gilbert,D.M., Groudine,M., Bender,M., Kaul,R., Canfield,T. *et al.* (2012) An encyclopedia of mouse DNA elements (Mouse ENCODE). *Genome Biol.*, **13**, 418.
4. ENCODE Consortium (2012) An integrated encyclopedia of DNA elements in the human genome. *Nature*, **489**, 57.
5. Bernstein,B.E., Stamatoyannopoulos,J.A., Costello,J.F., Ren,B., Milosavljevic,A., Meissner,A., Kellis,M., Marra,M.A., Beaudet,A.L., Ecker,J.R. *et al.* (2010) The NIH Roadmap Epigenomics Mapping Consortium. *Nat. Biotechnol.*, **28**, 1045.
6. Roider,H.G., Lenhard,B., Kanhere,A., Haas,S.A. and Vingron,M. (2009) CpG-depleted promoters harbor tissue-specific transcription factor binding signals—implications for motif overrepresentation analyses. *Nucleic Acids Res.*, **37**, 6305–6315.
7. Halperin,Y., Linhart,C., Ulitsky,I. and Shamir,R. (2009) Allegro: analyzing expression and sequence in concert to discover regulatory programs. *Nucleic Acids Res.*, **37**, 1566–1579.
8. Calo,E. and Wysocka,J. (2013) Modification of enhancer chromatin: what, how, and why? *Mol. Cell*, **49**, 825–837.
9. Conlon,E.M., Liu,X.S., Lieb,J.D. and Liu,J.S. (2003) Integrating regulatory motif discovery and genome-wide expression analysis. *Proc. Natl Acad. Sci. U.S.A.*, **100**, 3339–3344.
10. GuhaThakurta,D. (2006) Computational identification of transcriptional regulatory elements in DNA sequence. *Nucleic Acids Res.*, **34**, 3585–3598.
11. Thurman,R.E., Rynes,E., Humbert,R., Vierstra,J., Maurano,M.T., Haugen,E., Sheffield,N.C., Stergachis,A.B., Wang,H., Vernot,B. *et al.* (2012) The accessible chromatin landscape of the human genome. *Nature*, **489**, 75.
12. Boyle,A.P., Davis,S., Shulha,H.P., Meltzer,P., Margulies,E.H., Weng,Z., Furey,T.S. and Crawford,G.E. (2008) High-resolution mapping and characterization of open chromatin across the genome. *Cell*, **132**, 311–322.
13. Buenrostro,J.D., Wu,B., Chang,H.Y. and Greenleaf,W.J. (2015) ATAC-seq: a method for assaying chromatin accessibility genome-wide. *Curr. Protoc. Mol. Biol.*, **109**, 21–29.
14. Hashimoto,T., Sherwood,R.I., Kang,D.D., Rajagopal,N., Barkal,A.A., Zeng,H., Emons,B.J., Srinivasan,S., Jaakkola,T. and Gifford,D.K. (2016) A synergistic DNA logic predicts genome-wide chromatin accessibility. *Genome Res.*, **26**, 1430–1440.
15. Lee,D., Karchin,R. and Beer,M.A. (2011) Discriminative prediction of mammalian enhancers from DNA sequence. *Genome Res.*, **21**, 2167–2180.
16. Yang,B., Liu,F., Ren,C., Ouyang,Z., Xie,Z., Bo,X. and Shu,W. (2017) BiRen: predicting enhancers with a deep-learning-based model using the DNA sequence alone. *Bioinformatics*, **33**, 1930–1936.
17. 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 Res.*, **26**, 990–999.
18. Kelley,D.R., Reshef,Y.A., Bileschi,M., Belanger,D., McLean,C.Y. and Snoek,J. (2018) Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome Res.*, **28**, 739–750.
19. Benner,P. (2021) Computing leapfrog regularization paths with applications to large-scale *k*-mer logistic regression. *J. Comput. Biol.*, **28**, 560–569.
20. Benner,P. and Vingron,M. (2020) ModHMM: a modular supra-Bayesian genome segmentation method. *J. Comput. Biol.*, **27**, 442–457.
21. Ernst,J. and Kellis,M. (2012) ChromHMM: automating chromatin-state discovery and characterization. *Nat. Methods*, **9**, 215.
22. Mammana,A. and Chung,H.-R. (2015) Chromatin segmentation based on a probabilistic model for read counts explains a large portion of the epigenome. *Genome Biol.*, **16**, 151.
23. Schmidt,M., Le Roux,N. and Bach,F. (2017) Minimizing finite sums with the stochastic average gradient. *Math. Program.*, **162**, 83–112.
24. Defazio,A., Bach,F. and Lacoste-Julien,S. (2014) SAGA: a fast incremental gradient method with support for non-strongly convex composite objectives. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems. Series NIPS'14*. MIT Press, Cambridge, MA, USA, Vol. **1**, pp. 1646–1654.
25. Lee,D. (2016) LS-GKM: a new gkm-SVM for large-scale datasets. *Bioinformatics*, **32**, 2196–2198.
26. Ramsköld,D., Wang,E.T., Burge,C.B. and Sandberg,R. (2009) An abundance of ubiquitously expressed genes revealed by tissue transcriptome sequence data. *PLoS Comput. Biol.*, **5**, e1000598.
27. Gardiner-Garden,M. and Frommer,M. (1987) CpG islands in vertebrate genomes. *J. Mol. Biol.*, **196**, 261–282.
28. Shrikumar,A., Prakash,E. and Kundaje,A. (2019) GkmExplain: fast and accurate interpretation of nonlinear gapped *k*-mer SVMs. *Bioinformatics*, **35**, i173–i182.
29. Xi,W. and Beer,M.A. (2018) Local epigenomic state cannot discriminate interacting and non-interacting enhancer–promoter pairs with high accuracy. *PLoS Comput. Biol.*, **14**, e1006625.
30. Roider,H.G., Kanhere,A., Manke,T. and Vingron,M. (2006) Predicting transcription factor affinities to DNA from a biophysical model. *Bioinformatics*, **23**, 134–141.
31. Manke,T., Roider,H.G. and Vingron,M. (2008) Statistical modeling of transcription factor binding affinities predicts regulatory interactions. *PLoS Comput. Biol.*, **4**, e1000039.
32. Gelman,A. (2008) Scaling regression inputs by dividing by two standard deviations. *Stat. Med.*, **27**, 2865–2873.
33. Bailey,T.L., Boden,M., Buske,F.A., Frith,M., Grant,C.E., Clementi,L., Ren,J., Li,W.W. and Noble,W.S. (2009) MEME SUITE: tools for motif discovery and searching. *Nucleic Acids Res.*, **37**, W202–W208.
34. Desjardins,C.A. and Naya,F.J. (2016) The function of the MEF2 family of transcription factors in cardiac development, cardiogenomics, and direct reprogramming. *J. Cardiovasc. Dev. Dis.*, **3**, 26.
35. Sugawara,A., Sanno,N., Takahashi,N., Osamura,R.Y. and Abe,K. (1997) Retinoid X receptors in the kidney: their protein expression and functional significance. *Endocrinology*, **138**, 3175–3180.
36. Chaudhry,A.Z., Lyons,G.E. and Gronostajski,R.M. (1997) Expression patterns of the four nuclear factor I genes during mouse embryogenesis indicate a potential role in development. *Dev. Dyn.*, **208**, 313–325.

37. Gao,Z., Ure,K., Ables,J.L., Lagace,D.C., Nave,K.-A., Goebbels,S., Eisch,A.J. and Hsieh,J. (2009) Neurod1 is essential for the survival and maturation of adult-born neurons. *Nat. Neurosci.*, **12**, 1090–1092.
38. Li,Z., Schulz,M.H., Look,T., Begemann,M., Zenke,M. and Costa,I.G. (2019) Identification of transcription factor binding sites using ATAC-seq. *Genome Biol.*, **20**, 45.
39. Wang,K., Zhang,H., Ma,D., Bucan,M., Glessner,J.T., Abrahams,B.S., Salyakina,D., Imielinski,M., Bradfield,J.P., Sleiman,P.M. *et al.* (2009) Common genetic variants on 5p14.1 associate with autism spectrum disorders. *Nature*, **459**, 528–533.
40. Inoue,Y.U. and Inoue,T. (2016) Brain enhancer activities at the gene-poor 5p14.1 autism-associated locus. *Sci. Rep.*, **6**, 1–12.