



- (51) **International Patent Classification:**
G01N 33/53 (2006.01) **G06F 19/24** (2011.01)
G06F 19/18 (2011.01)
- (21) **International Application Number:**
PCT/US2016/032163
- (22) **International Filing Date:**
12 May 2016 (12.05.2016)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
62/160,079 12 May 2015 (12.05.2015) US
- (71) **Applicant:** THE JOHNS HOPKINS UNIVERSITY
[US/US]; 100 N. Charles Street, 5th Floor, Baltimore, MD
21201 (US).
- (72) **Inventors:** BEER, Michael; 327 Regester Avenue, Bal-
timore, MD 21212 (US). LEE, Dongwon; 3320 Sonia
Trail, #205, Ellicott City, MD 21043 (US).
- (74) **Agents:** BROWN, Charley, F. et al.; Ballard Spahr, LLP,
999 Peachtree Street, Suite 1000, Atlanta, GA 30309 (US).

(81) **Designated States** (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,
BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM,
DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR,
KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG,
MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM,
PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC,
SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** (unless otherwise indicated, for every
kind of regional protection available): ARIPO (BW, GH,
GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ,
TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU,
TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE,
DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU,
LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK,
SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ,
GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

[Continued on next page]

(54) **Title:** METHODS, SYSTEMS AND DEVICES COMPRISING SUPPORT VECTOR MACHINE FOR REGULATORY SE-
QUENCE FEATURES

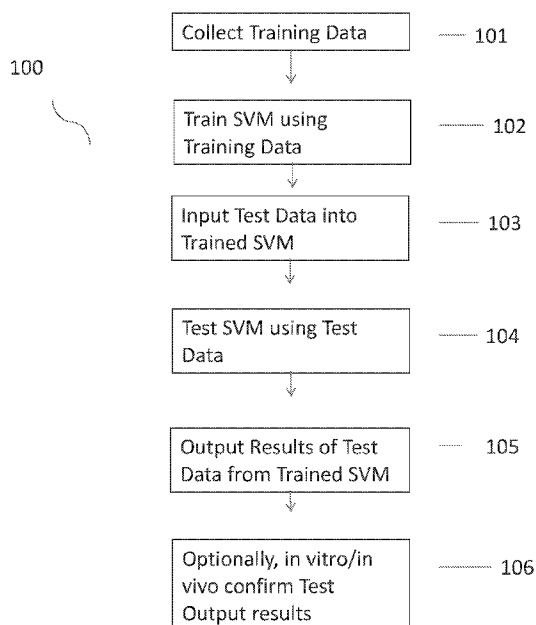


Fig. 1

(57) **Abstract:** The present disclosure comprises methods and systems for identifying variant predictive sequences in DNA, for example, in mammalian genomes. The variant predictive sequences are identified using a trained support vector machine (SVM) as disclosed herein, and such variant sequences can be used to diagnose disease and pathologies in a subject. With a diagnosis, the subject can be treated appropriately.



— with sequence listing part of description (Rule 5.2(a))

METHODS, SYSTEMS AND DEVICES COMPRISING SUPPORT VECTOR MACHINE FOR REGULATORY SEQUENCE FEATURES

CROSS-REFERENCE TO RELATED APPLICATIONS

[1] This application claims the benefit of U.S. Provisional Application No. 62/160,079, filed on May 12, 2015, which is incorporated herein by reference in its entirety.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH

[2] This invention was made with government support under HG007348 awarded by the National Institute of Health. The government has certain rights in the invention.

REFERENCE TO SEQUENCE LISTING

[3] The Sequence Listing submitted May 12, 2016 as a text file named “36406_0003P1_Sequence_Listing.txt,” created on May 11, 2016, and having a size of 2,004 bytes is hereby incorporated by reference pursuant to 37 C.F.R. § 1.52(e)(5).

FIELD OF THE DISCLOSURE

[4] Disclosed herein are methods, systems and compositions relating to computer-implemented methods for identifying nucleic acid regulatory sequences comprising methods and systems that comprise a support vector machine classifier.

BACKGROUND

[5] Gene regulatory sequences, such as enhancers, can control cellular activities, such as transcriptional activities, at a distance, independent of their position and orientation with respect to affected genes (Banerji 1981). For example, enhancer activity is modulated by interactions between sequence specific DNA binding proteins and sequence elements in the enhancer. Since individual transcription factor binding sites (TFBSs) can be relatively short and degenerate, TFBSs tend to be clustered to achieve precise temporal and developmental specificity (Kadonaga 2004). Factors bound to these sequences often interact with common coactivators, which, in turn, recruit the basal transcription machinery (Blackwood and Kadonaga 1998; Carter et al. 2002).

[6] Identifying the sequence elements and the combinatorial rules that determine enhancer function is necessary to fully understand how enhancers direct the spatial and temporal regulation of gene expression. Experimentally identified enhancers with similar functions can be a good starting point for in-depth study of the underlying rules encoded in the regulatory DNA sequence. However, the systematic functional identification of such enhancers has been limited due to the fact that they are often distant from the genes they regulate, requiring the interrogation of large amounts of potential regulatory sequence.

[7] What is needed are methods and systems for identifying regulatory sequences.

SUMMARY

[8] Disclosed are computer implemented systems and methods for enhancing knowledge discovered from data using a learning machine in general and a support vector machine in particular. In particular, the present disclosure comprises methods of using a learning machine for identifying regulatory sequences such as those that provide direction or control for cellular transcription, such as enhancers, repressors and/or insulators.

[9] In an exemplary embodiment, a system is provided for identifying regulatory sequences such as enhancer sequences in DNA from data using a support vector machine (SVM). An exemplary system comprises a storage device for storing a training data set and a test data set, and a processor for executing a support vector machine. The processor is also operable for collecting the training data set from the database, training the support vector machine using the training data set, collecting the test data set from the database, operating the trained support vector machine with the test data set, and identifying the sequences that are enhancer sequences by the trained SVM. Steps, such as *in vivo* or *in vitro* testing of the identified sequences to function as regulatory sequences, such as enhancer sequences, may be performed. An exemplary system may also comprise a communications device for receiving the test data set and the training data set from a remote source. In such a case, the processor may be operable to store the training data set and the test data set in a storage device. The exemplary system may also comprise a display device for displaying the test data results. The processor of the exemplary system may further be operable for performing each additional function described above. The communications device may be further operable to send a computationally derived alphanumeric classifier or other SVM-based raw output data to a remote source.

[10] The disclosure may further comprise providing the SVM for others to use, such as providing an Internet-based SVM that may or may not be trained, for use by others.

[11] Disclosed herein is a training data set of enhancer sequences as described herein.

[12] Disclosed herein is an algorithm used in training the SVM.

[13] Disclosed herein are enhancer sequences identified by the trained SVM.

[14] Disclosed herein are sequence-based computational methods and systems to predict the effect of regulatory variation using a classifier (gkm-SVM) which encodes cell-specific regulatory sequence vocabularies.

BRIEF DESCRIPTION OF THE FIGURES

[15] The accompanying figures, which are incorporated in and constitute a part of this specification, illustrate several aspects and together with the description serve to explain the principles of the disclosure.

[16] Figure 1 is a flowchart showing an overview of a method of the present disclosure.

[17] Figure 2 provides an overview of the methodology. (A) k-mer frequencies were calculated for each of the EP300-bound and negative genomic training sequences. These feature vectors ($x_1, \dots, x_n$) were used to find SVM weights, w , which most accurately separate the positive (enhancer) and negative (genomic) training sets. (B) These weights were used to predict genome-wide enhancers (light green), based on their SVM score. (Brown) positive, (blue) negative. A well-studied region around $DLx1$ and $DLx2$ was shown here, both known to be expressed in the forebrain. While the predicted enhancers often overlapped the training EP300 set (blue), novel enhancers were also predicted and often identified previously experimentally verified enhancers (red) absent from the EP300 training set. The predicted enhancers also preferentially occurred in conserved nonexonic regions (dark green) and regions enriched in EP300 signal (dark blue).

[18] Figure 3 shows classification results on each tissue-specific enhancer set. (A) Classification of forebrain enhancers vs. random genomic sequences. (B) Classification of midbrain enhancers vs. random genomic sequences. (C) Classification of limb enhancers vs. random genomic sequences. Each graph in A, B, and C compared an SVM trained on the full

set of 6-mers (solid), the top 100 selected 6-mers (dashed), and an alternative Naive Bayes classifier (dotted). Each curve was an average of five cross-fold validations on a reserved test set; error bars denote one standard deviation over the five cross-fold validation sets. Numbers in parentheses indicate the area under each ROC curve (auROC) for overall comparison. Both the full SVM and SVM with selected features performed very well and significantly better than Naive Bayes. Individually, each tissue-specific set can be accurately discriminated from nonenhancer genomic sequences. (D) Classification of specific tissues vs. other tissues. Forebrain (fb) and midbrain (mb) can be accurately discriminated from limb (lb) but not from each other (fb vs. mb), indicating common or overlapping modes of regulation. (E) Classification ROC curves for forebrain enhancers vs. random genomic sequences for larger negative set sizes. (F) Precision-recall curves for forebrain enhancers vs. random sequences corresponding to the ROC curves and negative sets in E; numbers in parentheses are auPRC. (G) Classification of EP300 forebrain enhancers, neuronal stimulus-dependent enhancers (CREBBP neuron), and mouse embryonic stem cell enhancers (EP300 ES) vs. random genomic sequence. Although the embryonic stem cell data set is somewhat less accurately classified, these SVMs successfully discriminated EP300 or CREBBP bound regions from random sequences. (H) Classification of EP300 fb, CREBBP neuron, and EP300 ES data sets vs. each other was also robust.

[19] Figure 4 shows predictive SVM sequence features were more conserved. Scatter plot between SVM weights and conservation scores (phastCons scores) for 6-mers in forebrain enhancers was shown. Two well-known TFBS, TAAT cores (red rectangles), and E-box elements (blue triangles) were highlighted. Three standard deviations above the mean (corresponding to P-value of ~ 0.001) was denoted for each axis independently. The sequence of all 6-mers beyond three standard deviations above the mean was displayed.

[20] Figure 5 shows predictive SVM sequence features were spatially clustered and distributions of minimum pairwise distances between the most predictive sequence features in forebrain enhancers vs. random genomic sequences. Ten 6-mers with the largest positive SVM weights (Table 1) were used. To measure the significance of these differences, 100 distinct full negative genomic sequence sets were generated (using the null model; disclosed herein). Each negative set had the same length, repeat fraction, and number of sequences as the EP300 forebrain enhancer training set. The predictive elements were significantly clustered in the forebrain enhancers compared to the random genomic sequences (the red

distribution is significantly shifted toward smaller minimum distance). At higher resolution (inset), distinct peaks around 11 bp, 22 bp, etc., were observed, suggesting positioning in phase with the periodicity of the DNA helix. P-values were indicated: (*) <0.01, (**) <0.001, (***) <0.0001.

[21] Figure 6 shows SVM-predicted regions were hypersensitive to DNase I in the relevant context. To independently confirm predictions with DNase I measurements in the embryonic mouse brain, the distributions of the average intensity of DNase I hypersensitivity of different forebrain SVM scoring regions were plotted. (A) DNase I hypersensitivity measured in E14.5 wholebrain. (B) DNase I hypersensitivity measured in an adult 8-wk kidney, as a negative control. Significant enrichments were observed only in high-scoring SVM-predicted regions in the brain.

[22] Figure 7 shows SVM-predicted enhancers were preferentially located near transcript start sites (TSSs) of forebrain-expressed genes. Here, plotted were the distribution of the distance between the EP300 and SVM predicted regions and the nearest forebrain-expressed gene [as assessed by the microarray experiments of Visel et al. (2009)]. Any region which overlapped a training set region was excluded from the analysis. Both the EP300 (red) and SVM-predicted regions were preferentially located within 10 kb of the TSS of a forebrain-overexpressed gene (above the axis). This was true whether a cut-off of SVM > 1.5 (green) or a more restrictive SVM > 2.0 (blue) was used to define the enhancer set. As a null set, the average of 100 randomized genomic positions, with a 95% confidence interval shown (gray) was compared. Interestingly, when calculating the same distributions for the distance between a EP300 or SVM predicted region and the nearest forebrain-underexpressed gene (below the axis), only the SVM predicted regions showed significant clustering toward the TSS, relative to the randomized control. Although the EP300 data preferentially identified activating enhancers in the forebrain, the SVM may have been detecting common sequence features shared in enhancers, which were repressive in the forebrain but were activating in other contexts.

[23] Figure 8 shows Table 1, Predictive 6-mers of EP300 forebrain.

[24] Figure 9 shows Table 2, Precision and sensitivity of detecting DNase I hypersensitive enhancers.

[25] Figure 10A and B shows a comparison of the performance of SVM models with different kernels and k-mer lengths, and a Naïve bayes classifier, using Visel's data set. ROC curves are shown for each of the three mouse tissues. Each curve is an average of 5 cross-fold validations on a reserved test set, and error bars denote one standard deviation over the 5 cross-fold validation sets. The numbers in the parenthesis indicate the average of the area under ROC curves (auROC). Three different lengths of k-mers, k=3, 5, 7, were tested. Generally, larger k exhibits better performance in terms of auROCs with some exceptions caused by over-fitting. (A) Using the full set of k-mers, SVM Classification results with three different kernels (Spectrum, Mismatch, and Gaussian) and Naïve Bayes classification results are shown. SVMs outperform Naïve Bayes classifiers in every case but one which failed to converge (SVM with 3-spectrum kernel on Midbrain). (B) Using only selected 6-mers, results of SVMs with spectrum kernels are presented. For each classification, a half of N 6-mers with the largest positive SVM weights and a half of N 6-mers with the largest negative SVM weights were selected (N=40, 100 and 200).

[26] Figure 11 A and B show graphs of length distribution and repeat fraction distribution between enhancers and random genomic sequences matched to EP300 enhancer set. For the null-sequence model, random sequences from the genome were selected to match the repeat fraction and length distribution of the sequences in the EP300 data set. The combined set of all Visel's EP300 bound regions are shown in red (the righthand bar), and the null sequence set is shown in blue (the lefthand bar).

[27] Figure 12 A-L are graphs showing the comparison between ROC curves and prevision-recall curves with larger negative sets. The scaling of negative set size is compared for all comparisons of positive sets vs. random genomic sequence for the 6-mer spectrum kernel SVM (Table 4, Figure 24). The genomic ratio of enhancers to non-enhancer sequence is very large (it is estimated that enhancers comprise 1-2% of the genome), so three negative sets (1x; 50x larger; and 100x larger than the positive enhancer set) were used for each case. The area under the ROC curve (auROC) or the area under the precision-recall curve (auPRC) is shown in parentheses. For large negative set size, auPRC is a more reliable measure of performance than the auROC curve, which is independent of negative set size, as expected.

[28] Figure 13 A and B show comparison between frequencies and SVM weights of k-mers. While the SVM features which are assigned large positive weights are generally over-represented in the EP300 bound regions relative to background genomic sequence, there was

not a strictly direct correlation between SVM weights and k-mer frequencies. (A) k-mer frequency in forebrain vs. SVM weights. (B) Normalized frequency difference between forebrain and random sequences, $\Delta f = (\text{freq}(\text{fb}) - \text{freq}(\text{rand})) / (\text{freq}(\text{fb}) + \text{freq}(\text{rand})) / 2$.

[29] Figure 14 shows average EP300 ChIPseq read coverage in the SVM predicted regions. In the graph shown, the 1% predicted is the highest/top line (at the 0 distance point), the 1% predicted without training is the middle line (at the 0 distance point), and the 1% random is the lowest/bottom line (at the 0 distance point). EP300 reads were significantly enriched in the SVM predicted regions: The middle point of the top 1% SVM predicted regions in forebrain were aligned at 0bp, the sequence around each peak was extended +/- 10kb in each direction, and the average coverage of EP300 reads in the surrounding regions is shown. Significant enrichments compared to random genomic sequence (by about two fold) is observed even after those regions which overlap with the original training set are excluded. This is further evidence that the SVM predicted regions which are not in the EP300 positive test set are in fact EP300-bound.

[30] Figure 15 A and B show correlation of SVM predictions and EP300 read density for genome wide scan. The correlation of SVM score and EP300 read density for all 1kbp regions across the genome is shown. (A) are all regions that partially overlap any positive training set region. (B) are all other genomic regions. The cloud of points with $\text{EP300} > 2$ ($\log_{10} 2 = 0.301$) and SVM score > 1 are the predicted enhancers, and it was expected that about 50% of these were true positive enhancers. Most regions with $\text{EP300} > 3$ are be in the positive training set, and are in (A) by construction. The regions in (A) with $\text{EP300} < 3$ are genomic 1kbp chunks which partially overlap a positive training set region.

[31] Figure 16 A and B show distribution of SVM scores for varying negative set size. In the graphs A and B, the line starting farthest to the left is the negative set, the line starting to the first line's left is the positive set. The distributions of SVM scores for negative set size (N=4000) and (N=120,000) are shown. While there was a shift in the (arbitrary) scale, the distributions were very similar, reflecting the fact that auROC is similar for N=4000, N=120,000, or N=240,000 negative sequences. On the other hand, as the negative set size increases, auPRC drops, because the higher scoring tail of the negative sequence score distribution becomes comparable to the bulk distribution of the positive sequences.

[32] Figure 17 shows the correclation between SVM scores from two separately trained SVMs. To investigate the robustness of the top SVM scoring regions, separate SVMs were trained using independently sampled random negative sequence sets, and compared the top SVM scoring regions using these different negative sequence sets. While there is some variation between the top scoring regions from different negative sets, only rarely do high scoring regions in one SVM not score highly the other SVMs, indicating that the predictions are robust to different realizations of the negative set. As shown in Table 5, Figure 25, there is 64.5% overlap between the top 1.0% regions for “Set1” and “Set2” SVMs, but 84.5% and 92.2% of the top 1% sites in Set1 are found in the top 2% and 3% of Set2, respectively. This graph compares the scores of chromosome 1 regions (to reduce the number of plotted points) from these two SVMs, showing very high correlation ($C=0.915$).

[33] Figure 18 shows classification of human homologous regions of the EP300 mouse training set. SVMs can discriminate human homologous EP300 bound regions from human random sequence. A positive human test set was generated by sequence alignment of the mouse EP300 training set regions to the human genome, varying the stringency for assigning homologous regions (70% identical, 90% identical, and 95% identical). As shown in the figure, all three of these sets can be classified with high accuracy (auROC=0.87, 0.88, 0.89), and classification power is relatively unaffected by the cut-off for determining homologous regions, again demonstrating the robustness of the SVM predicted enhancers.

[34] Figure 19 shows SVM predictions at the human Otx2 locus. To further compare the predictions of the SVM trained on the mouse EP300 bound regions and the SVM trained on human homologous sequence, two SVMs (mmSVM and hgSVM) were used to score the human genome Otx2, which is known to play a role in forebrain development. The raw hgSVM and mmSVM scores were quite similar, and most of the predicted enhancers above the 1% threshold overlap. One of these enhancers has been experimentally verified to have enhancer activity (CR).

[35] Figure 20 is Table 3, showing 6-mer SVM scores across the SOX-2 POU5F1(OCT4)-NANOG. Many large weight k-mers from the SVM trained on the EP300 ES dataset are subsequences that tile across the SOX2-OCT4 consensus oligo. The SOX2-OCT4-NANOG sequence, CATTGTYATGCAAAT, is SEQ ID NO:2.

[36] Figure 21 A-D shows graphs of PWMs vs k-mers as feature sets on forebrain and ZNF263. The figure shows comparisons of SVM performance using k-mers to an SVM using 811 known PWMs as features using ROC (A,C) and P-R curves (B,D). (A) On the forebrain enhancers, the k-mer SVM was more accurate than known PWMs alone, but a combination of k-mers and PWMs performed slightly better. (B) These differences in auROC translated to a dramatic reduction in auPRC for PWMs relative to k-mers only or combined k-mers and PWMs. (C) The k-mer SVM predicts ZNF263 bound regions from ChIP-seq with high accuracy (auROC=0.94), but the 811 PWM SVM is less accurate (auROC=0.83). (D) Again the lower auROC for PWMs corresponds to a significant decrease in auPRC for PWMs on the ZNF263 data (0.14 vs. 0.51), and a much higher false discovery rate.

[37] Figure 22 is a graph showing classifications using one negative set shared between different data sets. When training the SVMs for the three data sets (EP300 forebrain, CREBBP neuron, and EP300 ES), independent negative sets were used. To ensure that the predictive k-mers with large negative weights reflect their absence in the positive training set, not presence in various negative set realizations, one common negative set shared between the three data sets was generated. Since the length distribution and repeat fractions of the three data sets are different, the length of the positive sets was modified to be able to generate a single appropriate negative set. For Chen's and Kim's dataset, a fixed length was extended from the peaks reported. 800bps (+400bp from the peaks) was chosen to match with the lengths of forebrain data set as closely as possible (mean length of the forebrain data set is 816bp). The fixed 800bp length was chosen for the negative set because forebrain data set was relatively unaffected by the length distribution. 20000 random genomic sites for the negative set were sampled. To deal with the unbalanced positives and negative set sizes, the class weights were optimized for the positive sequences, and report the best result of each case. This figure shows the ROC curves of three different dataset classifications against the common negative set. This result is comparable to the original analysis. Table 11, Figure 31A and B show the top 15 positive 6-mers and top 10 negative 6-mers of each dataset from this analysis, which largely overlaps the results from the independent random negative sets, as shown in Table 1, Figure 8, Table 8, Figure 28 and Table 9, Figure 29.

[38] Figure 23 is a graph showing auROC and BEP using single chromosomes as test sets in 20-fold cross validation. To show that the cross validation procedure does not impact classification performance, SVMs were trained using 20-fold cross validation with test sets

consisting of all elements on a single chromosome (1-19, and X=20), instead of 5-fold cross validation used in the main text. The variation in auROC and Precision at the break-even point (BEP, where precision equals recall) is consistent with the varying size of the test sets. No chromosome is significantly more or less accurately predicted than the others.

[39] Figure 24 is Table 4, showing an outline of several analyses disclosed herein.

[40] Figure 25 is Table 5, showing further quantifying of the similarity of the predictions from the mouse and human SVMs, Figure 25 Table 5 shows the overlap of the top SVM scoring regions of the two SVMs. The mouse SVM (Set1) uses the mouse EP300 training set as positives and mouse random genomic regions as negatives, and the human SVM (Set2) uses human homologous regions of the mouse EP300 training set as positives and human random genomic regions as negatives. One third of top 1% scoring regions of Set1 are also found in the top 1% scoring regions of Set2. This overlap was quite significant considering the fact that the two SVMs were trained (learned) on different genomes.

[41] Figure 26 is Table 6, showing human enhancer prediction using a mouse vs. a human SVM.

[42] Figure 27A and B are Table 7, which (A) shows EP300-bound regions in each tissue of mouse embryo vs CREBBP peaks in activated cultured neurons; and (B), shows EP300 bound regions in each tissue of mouse embryo vs EP 300 peaks in embryonic stem cells. The significance of the overlap between Visel's EP300 bound regions and two other data sets were assessed: EP300 bound regions in ES cells and CREBBP bound regions in activated neurons. The number of EP300/CREBBP ChIP-seq peaks were counted in the new data sets which are located within the regions of Visel's data set (EP300 forebrain, midbrain, and limb), and calculated the p-value of the overlap. For a null hypothesis, it was assumed that the observed peaks could have been detected anywhere in potential regulatory regions, which were estimated as roughly 3.5% of entire genome (Waterston et al. 2002). Then the p-value of the overlap was calculated from the binomial distribution.

[43] Figure 28A and B is Table 8 showing predictive 6-mers of CREBBP Neuron, where (A) shows fifteen 6-mers with the largest positive SVM weights and (B) shows five 6-mers with the largest negative SVM weights.

[44] Figure 29A and B is Table 9 showing predictive 6-mers of embryonic stem cells, (A) fifteen 6-mers with the largest positive SVM weights, and (B) five 6-mers with the largest negative SVM weights.

[45] Figure 30 A and B are Table 10, showing a comparison of Predictive k-mers from the different data sets, (A) shows fifteen 6-mers with the largest positive SVM weights, and (B) shows fifteen 6-mers with the largest negative SVM weights.

[46] Figure 31 A and B are Table 11 showing predictive k-mers of three different datasets using common random negative sequences, (A) shows fifteen 6-mers with the largest positive SVM weights, and (B) shows fifteen 6-mers with the largest negative SVM weights.

[47] Figure 32 shows a workflow canvas for an exemplary method of the present disclosure. Shown are three different components from the kmer-SVM method disclosed herein, 'Generate Null Sequence', 'Train SVM' and 'Plot ROC Curve' and one optional module, 'Extract Genomic DNA'.

[48] Figure 33A-D-shows kmer-SVM analysis of ESRRB-binding sites. (A) ROC and PR curves for a kmer-SVM trained on ESRRB-bound genomic loci in ES cells versus 10-fold larger random genomic sequence. Default parameters were used for this analysis; Kernel type=Spectrum, K=6, C=1, E=1e-5, PSW=auto. (B) ROC and PR curves for the ESRRB PWM scores. (C) The top five positive and negative 6 mers recover the ESRRB motif. (D) reported in Chen et al., (2008) Integration of external signaling pathways with the core transcriptional network in embryonic stem cells. Cell, 133, 1106–1117. The sequence shown by the ESRRB motif is DBYSAAGSTSAN (SEQ ID NO:3), wherein D can be G, T, or A, wherein B can be G, C, or T, wherein Y can be T or C, wherein S can be G or C, and wherein N can be any nucleotide.

[49] Figure 34A-C shows kmer-SVM analysis of GR-binding sites. (A) ROC and PR curves for a kmer-SVM trained on GR bound loci in 3134 cells and AtT20 cells versus 10-fold larger random sequence. Default parameters were used; Kernel type=Spectrum, K=6, C=1, E=1e-5, PSW=auto. (B) The 10 most positive and negative 6 mers from 3134 cells and AtT20 cells recover the previously reported GRBE, AP1, AML1, HNF3, TAL1 and NF1 motifs (C) from John et al. ((2011) Chromatin accessibility pre-determines glucocorticoid receptor binding patterns. Nat. Genet., 43, 264–268), and additional novel accessory factors: CREB, TEAD1 and ZEB1. The GRBE motif is RGACAGWGTCY (SEQ ID NO:4); the

HNF3 motif is AWRRYAAAYA (SEQ ID NO:5); and the NF1 motif is YWGRWSSWGCCA (SEQ ID NO:6). R can be G or A, W can be A or T, Y can be T or C, and S can be G or C.

[50] Figure 35 A-C shows kmer-SVM analysis of sequence determinants of cell-type-specific GR binding. (A) ROC and PR curves for a kmer-SVM trained on GR-bound regions in AtT20 cells (positive set) versus GR-bound regions in 3134 Cells (negative set). Default parameters were used; Kernel type=Spectrum, K=6, C=1, E=1e-5, PSW=auto. (B) The accessory factor binding sites, including ZEB1, TAL1, HNF3, AML1 and AP1, are sufficient to distinguish the distinct sets of GR-bound regions in these two cell lines. The GRBE element is now present in both sets, is not predictive in this context and therefore does not receive a large weight. (C) ZEB1 motif from JASPAR database (Sandelin,A., et al. (2004) JASPAR: an open-access database for eukaryotic transcription factor binding profiles. Nucleic Acids Res., 32, 91D–94D.) is shown.

[51] Figure 36 A-C shows kmer-SVM analysis of EWS-FLI-binding sites.(A) ROC and PR curves for a kmer-SVM trained on EWS-FLI-bound regions in EWS502 cells and HUVEC cells versus random genomic sequence. Default parameters were used; Kernel type=Spectrum, K=6, C=1, E=1e-5, PSW=auto. (B) The 10 most positive, negative 6 mers from EWS502 cells and HUVEC Cells include binding sites the previously reported ETS and AP1 accessory factors, and novel accessory factors TEAD1 and ZEB1. (C) ETS (FLI1) from UniPROBE (16) and TEAD1 motif from JASPAR database are shown. The TEAD1 motif is NRCATTCYWVBB (SEQ ID NO:7). N can be any nucleotide, R can be G or A, W can be A or T, Y can be T or C, V can be A, G, or C, and B can be G, C or T.

[52] Figure 37 shows kmer-SVM versus PWM scores. The kmer-SVM AUROCs (Y-axis) of the 467 ChIP-seq data sets are compared with the best PWM AUROCs (X-axis). Default parameters were used; Kernel type=Spectrum, K=6, C=1, E=1e-5, PSW=auto. In general, kmer-SVM is much more accurate than any single PWM with one exception; the CTCF PWM (circle within triangle).

[53] Figure 38 A-D shows EP300 and H3K4me1 ChIP-seq signature at melanocyte enhancers. (A, left) Schematic of chr15:78,984,500–79,034,500 (UCSC Genome Browser; mm9) showing Sox10 and previously characterized melanocyte enhancer Sox10 MSC#5. (Right) Detailed view of the region immediately surrounding Sox10 MSC#5

(chr15:79030709–79033709), showing ChIP-seq data for EP300 (green) and H3K4me1 (blue) in melan-a. Rectangles are ChIP-seq peaks, and colored vertical bars below peaks show density of ChIP-seq reads in 10-bp bins. Gray bars at the bottom of inset show the phastCons score (Euarchontoglires). (B) Same scheme as in A, but showing the interval chr7:94,575,283–94,662,322 containing the Tyr gene and previously characterized melanocyte enhancer Tyr DRE-15kb. Interval shown to the right is chr7:94655287–94658287. (C) Number of ChIP-seq reads for H3K4me1 (blue, left axis) and EP300 (green, right axis) in a 5-kb window around the summits of 3622 EP300 peaks (averaged in 100-bp bins). (D) Heatmap showing the number of H3K4me1 ChIP-seq reads in a 3-kb window around 3,622 EP300 peaks.

[54] Figure 39 A-E shows H3K4me1-flanked EP300 peaks have multiple characteristics of melanocyte enhancers. (A) Visual representation of an exemplary method to identify putative melanocyte enhancers. (B) Average phastCons score (vertebrate, mm9) in a 1.5-kb window around the summit of 2489 putative melanocyte enhancers. (C) Top four motifs enriched in sequences of putative enhancers, with corresponding E-values (enrichment P-value times number of motifs tested; calculated by DREME) and factors predicted to bind to these motifs. (D) Number of putative enhancers in 1-MB window (100-kb bins) around the TSS of 2000 genes with the most abundant transcripts in melan-a (dark red), 2000 genes with least abundant transcripts in melan-a (light red), and 2000 randomly selected genes (white; average of five sets with SD represented by error bars). (E) Similar analysis to D, but using a 10-kb window and 1-kb bins.

[55] Figure 40 shows Table 12, which shows Gene Ontology (GO) terms associated with genes proximal to putative melanocyte enhancers.

[56] Figure 41 shows EP300 peaks that overlap H3K4me1-flanked regions have distinct properties. (A) Percent of EP300 peaks that directly overlap an annotated TSS (UCSC Genes; [dark green] peaks that overlap H3K4me1-flanked regions; [light green] peaks that do not overlap H3K4me1-flanked regions). Data for Heart (C57bl/6 mouse tissue taken at 8 wk), mES (Mouse ES-Bruce 4), and GM12878 generated by ENCODE and modENCODE consortia. (B) Average number ChIP-seq reads per peak for Pol2 (top row), H3K4me3 (middle row), and CTCF (bottom row) in a 2-kb window around the summits of indicated EP300 peaks (H3K4me1-flanked indicated by fl and dark green; non-H3K4me1-flanked, n-fl and light green). Three columns show data from the heart, mES, and GM12878, respectively.

(C) EP300 ChIP-seq fold enrichment (determined by MACS) of EP300 peaks that overlap H3K4me1-flanked regions (fl; darker green), and EP300 peaks that do not overlap H3K4me1-flanked regions (n-fl; lighter green). Corresponding P-values calculated by two-tailed t-test. Numbers of peaks are as follows: melan-a fl, 2489; n-fl, 1133; heart fl, 3324; heart n-fl, 23,236; mES fl, 1258; mES n-fl, 20,062; GM12878 fl, 3404; and GM12878 n-fl, 6703.

[57] Figure 42 A and B show Putative melanocyte enhancers direct reporter expression in melan-a. (A) Fold increase in luciferase reporter expression directed by indicated sequence relative to promoter-only control (P; white bar). Gray bars show fold increase of randomly selected putative enhancers (numbered 1–50). N (orange bar) represents the average of 10 negative regions. (Error bars) SD of three biological replicates, except in the case of N, where error bars show the standard deviation of 10 different negative regions. Note the difference in scale between bottom panel (onefold to 10-fold by one) and top panel (10-fold to 115-fold by 10). (Dotted lines) 10-fold, fivefold, and threefold thresholds (top to bottom). (B) Box plot summarizing results of reporter assays for 10 negative regions (top, orange) and 50 putative enhancers (bottom, gray). $P = 9.564 \times 10^{-7}$ by two-tailed t-test. Four outliers in putative enhancer group not shown in box plot (nos. 14, 22, 27, 46).

[58] Figure 43 A-E are a chart (A) and graphs showing deltaSVM can accurately predict SNPs associated with DNaseI Hypersensitivity. (a) An example of a deltaSVM calculation using a known dsQTL SNP (rs4953223). (b) 10-mer gkm-SVM scores across the dsQTL locus containing rs4953223 are shown. Only the functional SNP produces dramatic changes in gkm-SVM scores. (c) Effect sizes of dsQTL SNPs from Ref. 13 are well correlated with their deltaSVM scores. (d-e) deltaSVM predicts dsQTLs with far greater accuracy than existing methods. Discriminative powers are compared between various methods using 50x larger control SNP set. (d) ROC curve. (e) Precision-Recall curve.

[59] Figure 44 is an overview of a deltaSVM method. [left] The first step in calculating deltaSVM is to train a gkm-SVM classifier using a positive training set of putative regulatory sequences (identified by DNase I hypersensitivity, for example) and a negative training set of matched negative control sequences. The gkm-SVM generates a regulatory sequence vocabulary – a weighted list of all possible 10-mers, in which each 10-mer receives an SVM weight that quantifies its contribution to the prediction of whether a given sequence has putative regulatory function, or not. [right] After training, this regulatory sequence

vocabulary can be used to score the predicted impact of any sequence variant on regulatory activity, as shown here for a single nucleotide substitution in a melanocyte enhancer of the *Tyrp1* enhancer.

[60] Figure 45 A-D are plots showing correlation of deltaSVM and dsQTL effect size drops with increasing distance between the dsQTL SNPs and the center of the associated DNase I sensitive regions. The original set of dsQTLs were defined as SNPs within ± 1000 bp of co-varying hypersensitive regions.¹³ We find that deltaSVM is only consistent with dsQTL effect size (beta) when we constrain the set of dsQTLs to be within 200bp of the modulated DHS region: (a) 0~50 (bp), (b) 50~200 (bp), (c) 200~500 (bp), and (d) 500~1000 (bp). This analysis is consistent with a local mechanism of action for dsQTLs.

[61] Figure 46A-C are plots showing deltaSVM is strongly positively correlated with dsQTL effect size, and positively or negatively correlated with eQTL effect size depending on the sign of the correlation of dsQTL and eQTL. Degner et al reported that 16% of the dsQTLs were also eQTLs, but that 30% of the eQTL dsQTLs were anti-correlated with the expression change. These predictions are consistent with this observation: (a) deltaSVM is always positively correlated with dsQTL effect size (beta), (b) but because eQTL beta and dsQTL beta are anti-correlated 30% of the time, (c) deltaSVM and eQTL beta are only correlated (positively and negatively) if we treat the activating dsQTLs (darker) and repressive dsQTLs (lighter) separately.

[62] Figure 47 A-D are plots showing bases predicted to reduce the activity of functional regions are evolutionarily constrained. The average deltaSVM scores were calculated for all 3 possible mutations at each within LCL GM12878 DHSs and the conservation (phyloP) for bases causing (a) negative, (b) neutral, and (c) positive deltaSVM predicted impact (the top 1% negative deltaSVM, 1% of deltaSVM near 0, and the top 1% positive deltaSVM, N=63,123) were compared. (d) Differential distributions relative to neutral deltaSVM bases. Both negative and positive deltaSVM bases are more conserved than neutral deltaSVM bases; $P < 1e-300$ (under machine precision) and $P < 1e-14$, respectively (Kolmogorov-Smirnov test). Also, negative deltaSVM bases are much more conserved than positive deltaSVM bases (average phyloP: 1.00 vs. 0.20, $P < 1e-300$).

[63] Figure 48 A-D are plots showing deltaSVM accurately predicts change in luciferase expression in targeted mutagenesis of *Tyr* and *Tyrp1* melanocyte enhancers. (a,b) Base by

base evaluation of all possible substitutions as scored by deltaSVM. Black circles mark substitutions that were tested in luciferase assays. (c,d) Correlation of deltaSVM prediction and observed normalized luciferase expression. Green circles indicate previously tested binding site^{20,21}. Error bar is one standard deviation of the changes in luciferase expression (4 biological replicates per variant).

[64] Figure 49 are plots showing deltaSVM accurately predicts change of expression in massively parallel reporter assays. (a) Correlations of deltaSVM predictions and observed in vivo mutation effect size in the ALDOB enhancer in mice²². (b) Correlation of deltaSVM and mutated enhancers in K562 cells²⁴. (c) Correlation of deltaSVM and mutated enhancers in HepG2 cells²⁴.

[65] Figure 50 is a plot showing correlations of deltaSVM and in vivo mutation effect size in the ALDOB enhancer using aggregate model. deltaSVM scores of all 3 possible mutations at each base were averaged, and compared the expression changes from univariate model reported by Patwardhan et al.

[66] Figure 51A-F are charts and a table showing that deltaSVM correctly identifies the causal validated SNP in previously studied GWAS loci associated with prostate cancer, fetal hemoglobin levels, and LDL cholesterol levels.

[67] Figure 52 is a plot showing that high confidence predicted causal SNPs in loci associated with autoimmune disease. The significance of the maximum of Abs (deltaSVM) depends on the number of flanking candidate causal SNPs. Sampling of random SNPs scored with the Th1 gkm-SVM yields the solid curves for the top 2% of all loci, and the mean, with standard deviation shown (dashed). 17 of the 413 immune associated loci exceed the 2% threshold, while 8 would be expected by chance.

[68] Figure 53A-D shows that gkm-SVM outperforms kmer-SVM over a wide range of k -mer length. Both gkm-SVM and kmer-SVM were trained on (A) CTCF bound and (B) EP300 bound genomic regions using different word lengths (k for kmer-SVM and l for gkm-SVM). The parameter k for gkm-SVM was fixed at 6. While AUCs of the kmer-SVMs show significant overfitting in both cases as k gets larger (dotted), gkm-SVMs accuracy is higher for a broad range of larger l (solid). Results using the truncated Gkm-SVM with $m_{\max} = 3$ are shown as dashed lines and AUCs of these faster approximations are comparable when the difference between $m_{\max}$ and $l - k$ are relatively small. ROC for the optimal k or l for each

case are shown in (C) and (D). Gkm-SVMs (solid) consistently outperform kmer-SVMs (dashed) on both data sets. Error bars here and below represent 5-fold CV standard deviation.

[69] Figure 54 A-C shows gkm-SVM consistently outperformed kmer-SVM and the best known PWM on human ENCODE ChIP-seq data sets. (A) gkm-SVM and kmer-SVM were trained on the complete set of 467 ENCODE ChIP-seq data sets (with $k = 6$ for kmer-SVM, and $l = 10$ and $k = 6$ for gkm-SVM). gkm-SVM AUC was consistently higher than kmer-SVM with only a few very minor exceptions. The gkm-SVM method specially outperformed the kmer-SVM for the data sets bound by members of the CTCF complex, highlighted as purple circles. (B) Also compared were gkm-SVM and the best known PWM on the same data sets, and gkm-SVM AUCs were significantly higher than the PWM AUC in almost all cases. (C) The ENCODE data sets were divided into four groups: (1) no PWM, (2) only one PWM, (3) two PWMs, and (4) three or more PWMs identified by Wang *et al.* Then, for each group except the first one, the number of PWMs recovered by the disclosed method was calculated. At least one PWM was recovered for more than ~90% of the data sets.

[70] Figure 55 A-B shows the comparison of gkm-SVM and existing methods on the mouse forebrain EP300 data set: (A) For each method, averages of 5-CV AUCs are shown as a function of the word length with the optimal number of mismatches, m , held fixed. Also shown are gkm-SVM results using fixed $k=6$ and varying $m_{\max}$. (B) Running time for each of the kernel computations shown in (A). Gkm-kernels show better classification performance and significantly more efficient computation at peak AUC.

[71] Figure 56A-B shows gapped k -mer features also improved performance of Naïve Bayes classifiers. Naïve-Bayes classifiers were trained on (A) CTCF bound and (B) EP300 bound genomic regions using different word lengths, k , using both actual k -mer counts (dashed), and estimated k -mer counts from the gkm-filter (solid). As shown above for SVM, the Naïve-Bayes accuracy as measured by AUC is systematically higher using gapped k -mer estimated frequencies instead of actual k -mer counts, further supporting the utility of gapped k -mer based features. For CTCF the Naïve-Bayes AUC is comparable to the best SVM (dotted red lines), but for EP300 the SVM outperforms the Naïve-Bayes classifier.

[72] Figure 57 shows fast computation of mismatch profiles using k -mer tree structure. As an example, $l=3$ and three sequences $S_1=AAACCC$, $S_2=AAAAA$, and $S_3=ACC$ were used

to build the k -mer tree. The leaves (nodes at depth $d=l=3$) correspond to 3-mers AAA, AAC, ACC, and CCC. The sequence ID and the number of times each 3-mer appeared in each sequence are stored for each leaf. Each node t_i at depth d represents a sequence of length d , denoted by $s(t_i)$, which is determined by the path from the root of the tree to t_i . For example, $s(t_2)=C$ and $s(t_4)=AC$. DFS is started at the root node, t_0 . When visiting each node t_i at depth d , we compute the list of all the nodes t_j at depth d for which $s(t_i)$ and $s(t_j)$ have at most m_{max} mismatches. Also computed were the number of mismatches between $s(t_i)$ and $s(t_j)$. When reaching a leaf, the corresponding mismatch profile $N_m(S_i, S_j)$ was incremented for each pair of sequences S_i in that leaf and S_j in the list.

[73] Figure 58 is an exemplary operating environment.

DETAILED DESCRIPTION

[74] The present disclosure provides methods, systems and computer programs for identifying regulatory sequences, for example enhancer sequences, repressor sequences and/or insulator sequences in nucleic acid sequences, using learning machines. For example, the present disclosure is directed to methods and systems for identifying enhancer sequences from DNA using a trained SVM (support vector machine) that provides information regarding known enhancer sequences. Though the description herein is directed to enhancer sequences, one of skill in the art is capable of using the methods described herein for identifying other sequences found in nucleic acid genomes.

[75] The present disclosure comprises a discriminative computational framework to detect regulatory sequences from DNA sequence alone that does not rely on conservation or known TF binding specificities. Methods comprise using a support vector machine (SVM) to differentiate enhancers from nonfunctional regions, using DNA sequence elements as features. SVMs (Boser et al. 1992; Vapnik 1995) have been successfully applied in many biological contexts (for review, see Schölkopf et al. 2004; Ben-Hur et al. 2008): cancer tissue classification (Furey et al. 2000); protein domain classification (Karchin et al. 2002; Leslie et al. 2002, 2004); splice site prediction (Rätsch et al. 2005; Sonnenburg et al. 2007); and nucleosome positioning (Peckham et al. 2007). For example, for identifying enhancer sequences, because of the potentially diverse mechanisms which direct EP300 and CREBBP binding, a complete set of DNA sequence features was used to capture combinations of binding sites active in different tissues and times of development. To study these distinct

modes of regulation, EP300/CREBBP binding in mouse embryos (Visel et al. 2009), activated cultured neurons (Kim et al. 2010), and embryonic stem (ES) cells (Chen et al. 2008) was investigated. Visel's data set was first used, where several thousands of EP300-bound DNA elements were collected by ChIP-seq in dissected mouse embryo forebrain, midbrain, and limb. A method was tested by predicting enhancers vs. random sequence and between EP300/CREBBP ChIP-seq data sets. These comparisons revealed a diversity of predictive sequence features, both within and across data sets. Table 3, Figure 24 provides an outline of the analyses performed.

[76] The present disclosure comprises computer-implemented systems and methods for systematically identifying functions regulatory variants in the genetic code and methods of diagnosing diseases or pathologies related to such variants.

PARAGRAPH FROM Gapped Kmers

[77] In general, the present disclosure comprises computer-implemented systems and methods for identifying nucleic acid sequence features, such as regulatory features or sequence variants that are predictive for disease or pathology, wherein the methods and systems comprise three main components: (i) generating positive and negative sequence sets, (ii) training the SVM classifier and (iii) analyzing its performance and predictive sequence features. In an aspect, a positive training sequence set may be provided by the user, and such data may be, for example, in the form of a BED file of coordinates or sequence data in FASTA format, including genomic coordinates. A negative sequence set may be generated by methods disclosed herein, for example as a 'Generate Null Sequence' module. SVM training was fairly transparent, and takes the positive and negative sequence sets as input and in an aspect, produces a set of k-mer weights and predicted class labels as output using cross-validation. Additional methods disclosed herein use gapped k-mer weights. The performance of the SVM classifier is summarized by Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves, and features are ranked by their significance. In an aspect, Figure 32 shows a general workflow and this workflow can also be used as a template for an exemplary analysis method and system of the present disclosure. In an aspect, Figure 44 shows a general workflow and this workflow can also be used as a template for an exemplary analysis method and system of the present disclosure.

Generation of sequence sets

[78] In a method of the present disclosure, a kmer-SVM classifier can use as training data a set of positive sequences provided by a user. Such positive data set may be for example, a FASTA file of positive sequences obtained through ChIP-seq, DNase-seq or another experimental assay. A negative sequence set may be provided by a user or may be generated as described herein. In an aspect, it is desirable that a SVM identifies sequence features specific to the positive regions, the GC content, length and repeat fraction is matched when constructing the negative set, otherwise sequence features could be predictive simply by their enrichment or absence in the biased negative set. A set of the three distributions of GC, length and repeats in the positive set are referred to herein as its 'sequence profile' and the Generate Null Sequence method in general matches this sequence profile for the negative set by using the following random sampling procedure. First, a positive sequence is randomly selected, and the same chromosome is sampled (examined) for a match in terms of length, GC content and repeat fraction, which does not overlap any positive sequence or existing negative sequences by even one base pair. This random selection process is then repeated until the negative set has reached a predetermined size. In an example, the random selection process used a pre-computed table of genomic indices, for example, those provided for the *Caenorhabditis elegans*, *Drosophila melanogaster*, mouse and/or human genome. A full negative sequence set then by construction closely approximates the sequence profile of the positive set. In some methods, a user can exclude regions other than the input positive sequences from consideration for negative sequence generation. A method of the present disclosure may comprise the use of a negative set which is larger than the positive set, as doing so may improve the statistical robustness of the classifier. A method of the present disclosure may comprise the use of a negative set which is smaller than the positive set. A user may specify (predetermine) the size of the negative set as an integral multiple of the number of positive sequences (e.g. 10x). As some positive sequences may not have exact matches in terms of GC content or repeat fraction, a user can specify the percentage of GC content or repeat fractions by which a generated null sequence may differ from its corresponding positive sequence. This additional flexibility speeds the generation of the negative set and affects how precisely the negative set sequence profile matches the positive set sequence profile. Also, distinct realizations of null sequence sets may be generated by varying the Random Number Seed parameter. In an example, the output of the Generate Null Sequence tool was a BED file that described the coordinates of the negative genomic

intervals.

[79] After the coordinates are specified, the actual sequences needed for SVM training are generated from the positive and negative coordinates, for example, which may be obtained in a BED file by a built-in Galaxy tool: ‘Fetch Sequences’, whose output may be FASTA format DNA sequence files.

SVM training

[80] An SVM is a classifier, which attempts to find a hyper-plane boundary in feature space that separates elements of the positive and negative sequence sets. SVMs use techniques known as ‘kernels’, which allows for defining similarities between any two data points without explicit mapping of the data into a higher-dimensional feature vector space. A set of kernels called ‘string kernels’ have been developed for analyses of sequence data sets and have achieved great success in computational biology. A Train SVM step may use a string kernel, for example, the spectrum kernel (Leslie, C. et al., (2002) The spectrum kernel: a string kernel for SVM protein classification. *Pac. Symp. Biocomput.*, 7, 566–575.). In an aspect, the features may be the complete set of k-mers, and their frequencies may be calculated from the input data (positive and negative sequence sets), such as that provided by FASTA files. The training method step, Train SVM, may comprise generating the normalized k-mer count vector for each sequence and then finding the SVM internal parameters (support vectors) that most accurately distinguished the positive and negative sets. Train SVM may comprise one or more kernels, for example, the spectrum kernel (using a single length k-mer) and/or the weighted spectrum kernel (using a user specified range of k’s, with equal weighting). In both cases, reverse complement k-mers may be treated as separate instances of the same feature. An example comprises using the SVM Shogun toolbox (Sonnenburg, S., et al., (2010) The SHOGUN machine learning toolbox. *J. Mach. Learn. Res.*, 11, 1799–1802.). A method step of training the SVM performs two tasks: it generates a set of ranked k-mer-SVM weights, and it generates a set of class predictions using CV. A given k-mer’s score can be thought of as a measure of the degree to which that k-mer contributes to the discriminatory power of the classifier. The weights may be output to a table, for example, labeled Weights.

CV

[81] As is standard in machine learning, CV may be used to assess classifier performance. The initial positive and negative sets may be randomly partitioned into n distinct sets (for n -

fold CV), and the ROC and PR performance of each test set may be generated using a classifier trained on the other $n-1$ sets. The number of CV sets is a parameter, which can be specified by the user. This may be repeated for all n partitions such that in the end each partition may be used for both training and test-set scoring. The result of this process may be the set of scores for test-set sequences in each round of CV, which may be output to a table, for example, labeled Predictions.

[82] An aspect of a method of the present disclosure comprises three parameters for SVM learning that may be adjustable (k , C and E). If the spectrum kernel is used, k specifies a single kmer length, whereas if the weighted spectrum kernel is used, minimum and maximum values for k must be set. Using a single k is somewhat easier to interpret in the beginning, as the vocabulary is simpler. Using a range of k values does have the advantage that similar kmers of slightly varying length and composition should all receive significant weights, increasing confidence in interpretation. Also, using a range (e.g. 5–8) usually performs incrementally better than a single k in terms of overall classification accuracy.

[83] The SVM maximizes the margin between the positive and negative sequences while simultaneously minimizing errors (sequences on the wrong side of the boundary). The relative importance of misclassification error is weighted by the regularization parameter, C . In practice, this affects over-fitting. A small C will result in less over-fitting of the SVM at the expense of slightly greater training classification error, whereas a large C will result in more over-fitting of the SVM. With unbalanced positive and negative set sizes, a user may want to use a separate regularization parameter for positive and negative sequences, reflecting the relative importance of errors. For example, as disclosed herein, methods may comprise using an additional parameter Positive Set Weight or PSW. In an example, the regularization parameter for the positive set was $C * PSW$, whereas for the negative set, it was C . The default setting was $PSW = 1 + \log(N/P)$, which weighed positives more heavily when the negative set was large. The rationale behind this formula appears to be that optimal PSWs usually follow the logarithm of the ratio between positives and negatives. In practice, results were insensitive to C and PSW, unless there was a significant imbalance between the positive and negative set sizes. The precision parameter E constrains the precision of the SVM classifier. Increasing E results in a reduced number of support vectors and can lead to a more robust classifier by reducing the requirements on the accuracy of the classifier on the training set. In practice, the results should be insensitive to the choice of E , and a default

value is adequate. Runtime may increase as a function of the total number of sequences in the positive and negative data sets, and may range from under 1 to 40 min, for example, see the data sets shown in Examples.

Interpretation of kmer SVM weights

[84] The output of SVM training may be a list of k-mer weights, and it is the weighted sum of normalized k-mer counts in a sequence that determines the predicted class. In biological terms, the presence of k-mers with large positive weights significantly increases a sequence's likelihood of being positive (e.g. being an enhancer or being bound by a TF in a specific cell type). Large negative weights are equally informative, as their absence significantly increases the probability of being positive (e.g. a binding site for a transcriptional repressor). For example, in a method the weights file output by the Train SVM step may list all k-mers and their corresponding scores. The SVM weight is a continuous valued quantity, and large absolute value is a direct measure of significance. It is the scores with large absolute values that will be of particular value to the biologist. The TFs binding the highest and lowest scoring k-mers, if previously studied, can be found using database matching programs such as TOMTOM, using the UniPROBE, TRANSFAC and JASPAR databases. Finding the best PWM match to a k-mer does not necessarily imply that that factor binds the k-mer in the given context because many TFs have overlapping specificities, and the PWM databases are far from complete. However, large positive scoring k-mers are often recognizable as TF-binding sites known to be important in the cell type of interest, whereas large negative scoring k-mers have identified an important role for repressors in previously unknown contexts (3).

Classification performance analysis

[85] The area under the ROC curve (AUROC) and the PR curve (AUPRC) are measures of the accuracy of the classifier. AUROC corresponds to the probability that a randomly selected positive sequence will score higher than a randomly selected negative sequence. For example, for each possible SVM score threshold, the true positive rate [$TPR = TP/(TP+FN)$, or sensitivity] and false positive rate [$FPR = FP/(FP+TN)$, or 1-specificity] at this threshold may be calculated, where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and the FN is the number of false negatives. The ROC curve plots TPR versus FPR. The PR curve plots Precision versus Recall, where,

Precision = TP/ (TP+FP) and Recall = TPR. A method of the present disclosure may comprise assessing the accuracy of the classifier, wherein assessing may comprise calculating ROC and/or PR curves, and/or AUROC and/or USPRC using the classifier output information.

[86] The ROC and PR curves are slightly different measures of the classification performance of the trained SVM: the ROC emphasizes true and false positive rates, whereas the PR curve emphasizes true positive predictions. This difference results in the ROC possibly overestimating the accuracy of a classifier for data sets with large imbalances in the positive and negative class sizes, as is typical of genomic predictions with large negative sets. The PR curve is more appropriate in the case of large negative sets, yielding more accurate evaluations of classifier performance because it directly assesses the accuracy of positive predictions.

[87] Sequence features in an experimentally identified enhancer set were sufficient to train the SVM to accurately discriminate enhancers from random genomic regions. Data presented herein shows that the most predictive sequence elements were related to biologically relevant transcription factor binding sites. Examples herein show that some sequence elements were significantly absent in the enhancers (those with large negative SVM weights). For example, binding sites for the zinc finger E-box binding homeobox (ZEB) transcription factor family was depleted in the forebrain enhancers, consistent with its biological role as a transcriptional repressor (Vandewalle et al. 2008). In addition, it was found that enriched sequence elements were positionally constrained within the enhancers, and that they were more evolutionarily conserved than less predictive elements in the enhancers, reflecting the combinatorial structure of tissue-specific enhancers.

[88] The SVM methods and systems of the present disclosure can predict putative enhancers in both the mouse genome and the human genome from DNA sequence alone. Many of these novel enhancers overlap with regions enriched in EP300 ChIP-seq reads, exhibit greatly increased hypersensitivity to DNase I in the mouse brain, and were proximal to biologically relevant genes. All of these assessments exclude the original EP300 training set enhancers from the analysis. The successful identification of tissue-specific DNase I hypersensitive sites provides powerful independent evidence for the validity of the method disclosed herein for identifying enhancer sequences.

[89] Most investigations make use of two complementary approaches to detect putative regulatory regions: comparative genomics, which identifies enhancers by their sequence conservation across related species; and functional genomics, which identifies enhancers by the common binding of transcriptionally associated factors or marks (for review, see Noonan and McCallion 2010). Comparative genomics is based on the generally accepted hypothesis that functionally important regulatory sequences are under purifying selection. As a result, conserved noncoding sequences (CNSs) are natural candidates for putative enhancers. Early studies used CNSs to detect putative enhancers and test their activity in zebrafish or mouse reporter assays (Woolfe et al. 2004; Pennacchio et al. 2006; Visel et al. 2008). Although these conservation- based approaches achieve some success, limitations also exist. The function and spatio-temporal specificity of CNSs cannot be determined by conservation alone and, therefore, requires additional experimentation. More importantly, several studies have shown that noncoding sequences that apparently lack conservation (as assessed by sequence alignment) may still contain functional regulatory elements (Fisher et al. 2006; ENCODE Project Consortium 2007; McGaughey et al. 2008).

[90] Functional genomics is an experimentally driven approach that utilizes recently developed techniques of microarray hybridization or massively parallel sequencing in combination with chromatin immunoprecipitation (ChIP) on specific transcription factors (Johnson et al. 2007; Robertson et al. 2007), chromatin signatures (Heintzman et al. 2007, 2009), or coactivators (Visel et al. 2009; Kim et al. 2010). Specifically, some chromatin signatures or coactivator association (such as monomethylation of lysine 4 of histone H3, acetylation of lysine 27 of histone H3, and binding by coactivators EP300/ CREBBP) are predictive markers of enhancer activity (Heintzman et al. 2007, 2009). The transcriptional coactivators EP300 (also known as P300) and CREBBP (also known as CBP) have proven to be useful for enhancer identification because of their general roles as cofactors in mammalian transcription. Through highly conserved protein-protein interactions, EP300/CREBBP are hypothesized to operate as coactivators in at least three ways: as a direct bridge between sequence-specific transcription factors (TFs) and RNA Polymerase II, as an indirect bridge between sequence specific TFs and other coactivators which recruit RNA Pol II, or by modifying chromatin structure via intrinsic acetyl-transferase activity (Chan and La Thangue 2001). Several studies have reported genome- wide mapping of EP300/CREBBP-bound enhancers in different contexts, for example, tissue- specific activity in dissected mouse tissue (Visel et al. 2009) and environment-dependent activity in neurons (Kim et al. 2010).

Visel et al. validated that 90% of the EP300 enhancers tested recapitulated the expected spatial and temporal activity in vivo in a transgenic mouse enhancer assay. Functionally identified EP300-bound regions, thus, provide a robust starting point for further investigation of enhancers and their sequence properties.

[91] In principle, a complete understanding of enhancer mechanism would include a description of specific internal sequence features and how they contribute to enhancer function. Previous studies that have attempted to predict enhancers from sequence have typically used sequence conservation, colocalization of previously characterized TFBSs [from databases such as TRANSFAC (Matys et al. 2003) or JASPAR (Bryne et al. 2008)], or a combination of the two. Many of these existing approaches were assessed by Su et al. (2010), who found that some were successful in identifying enhancers in *Drosophila* but that few generalized to mammalian systems. The most successful method in mammalian enhancer prediction used a combination of conservation and low-order Markov models of sequence features (Elnitski et al. 2003; King et al. 2005). In more recent work, Leung and Eisen (2009) used word frequency profile similarity between pairs of sequences to detect novel enhancers, but training on small numbers of enhancers can be susceptible to noise. Another notable recent computational approach uses combinations of known TFBSs and de novo position weight matrices (PWMs) to detect enhancers (Narlikar et al. 2010).

SVM Description

[92] Exemplary embodiments of the present disclosure will hereinafter be described with reference to the drawing, in which like numerals indicate like elements throughout the several figures. Figure 1 is a flowchart illustrating a general method 100 for identifying enhancer sequences using an SVM. The method 100 begins at collection of training data, step 101. Training data comprises a set of data points having known characteristics. Training data may be collected from one or more local and/or remote sources. The collection of training data may be accomplished manually or by way of an automated process, such as known electronic data transfer methods. Accordingly, an exemplary embodiment of the present disclosure may be implemented in a networked computer environment. As described herein, training data may comprise positive and negative sequence sets. Next, at step 102, the learning machine is trained using the training data. As is known in the art, a learning machine is trained by adjusting its operating parameters until a desirable training output is achieved. The determination of whether a training output is desirable may be accomplished either manually

or automatically by comparing the training output to the known characteristics of the training data. A learning machine is considered to be trained when its training output is within a predetermined error threshold from the known characteristics of the training data.

[93] At step 103, test data is input into the trained SVM. Test data may be optionally collected in preparation for testing the trained learning machine. Test data may be collected from one or more local and/or remote sources. In practice, test data and training data may be collected from the same source(s) at the same time. Thus, test data and training data sets can be divided out of a common data set and stored in a local storage medium for use as different input data sets for a learning machine. Then, at step 104, the learning machine is tested using the test data. At step 105, the output results of test data from the learning machine is examined to determine if the results are desirable, reliable, accurate, or whatever criteria is established for the results. Optionally, at step 106, the output results may be verified or confirmed by in vivo or in vitro tests to determine if the enhancer sequences identified function as enhancer sequences in one or more tissues at the same or different times during differentiation, growth, cell death, or other cellular life timepoints.

[94] An SVM implements a specialized algorithm for providing generalization when estimating a multi-dimensional function from a limited collection of data. An SVM may be particularly useful in solving dependency estimation problems. More specifically, an SVM may be used accurately in estimating indicator functions (e.g. pattern recognition problems) and real-valued functions (e.g. function approximation problems, regression estimation problems, density estimation problems, and solving inverse problems). The concepts underlying the SVM are explained in detail in a book by Vladimir N. Vapnik, entitled Statistical Learning Theory (John Wiley & Sons, Inc. 1998), which is herein incorporated by reference in its entirety. Accordingly, a familiarity with SVMs and the terminology used therewith are presumed throughout this specification.

[95] Support vector machines were introduced in 1992 and the "kernel trick" was described. See Boser, B, et al., in Fifth Annual Workshop on Computational Learning Theory, p 144-152, Pittsburgh, ACM which is herein incorporated in its entirety. A training algorithm that maximizes the margin between the training patterns and the decision boundary is provided herein. The techniques may be applicable to a wide variety of classification functions, including Perceptrons, polynomials, and Radial Basis Functions. The effective number of parameters may be adjusted automatically to match the complexity of the problem.

The solution may be expressed as a linear combination of supporting patterns. These are the subset of training patterns that are closest to the decision boundary. Bounds on the generalization performance based on the leave- one-out method and the VC-dimension are given herein. A memory-based decision system with optimum margin may be designed wherein weights and prototypes of training patterns of a memory-based decision function are determined such that the corresponding decision function satisfies the criterion of margin optimality. Methods of the present disclosure comprise use of one or more SVM to identify regulatory sequences, such as enhancer sequences from native DNA or DNA genomes. Data input or output from the one or more SVMs may be pre- or post-processed by methods known to those skilled in the art.

Enhancers can be accurately predicted from DNA sequence

[96] Methods and systems of the present disclosure comprise identifying which sequence features are specific to enhancers and investigating the degree to which functional enhancer regions in a mammalian genome using only DNA sequence features in these regions can be identified. Recent genome-wide experiments that identified EP300 binding sites by ChIP-seq (Visel et al. 2009) in three different tissues (forebrain, midbrain, and limb) at embryonic day 11.5 in mice were used. Cross-linking in dissected tissue at a particular time point during development can identify tissue-specific enhancers, even when the developmental regulators that mediate EP300 binding are unknown. While EP300 ChIP may not detect all the enhancers active under these conditions, this data set was used to identify sequence features responsible for EP300 binding in these tissues.

[97] To model DNA sequence features, a support vector machine framework was used. In brief, an SVM finds a decision boundary that maximally distinguishes two sets of data, here a positive (enhancer) and negative (random genomic) sequence set. The basic approach is outlined in Figure 2A, and full details are disclosed herein. Weights, w_i , determined the contribution of each feature to this boundary. Once the set of sequence features, x_i , was specified, the weights were optimized to maximize the separation between the two classes. Sequence features used were the full set of k-mers of varying length (3–10 bp). While other authors have successfully used databases of experimentally characterized TFBSs as sequence features (Gotea et al. 2010), because the binding specificity of many transcription factors (TFs) has yet to be determined, in the present disclosure, k-mers (oligomers of length k) were used as sequence features because they are an unbiased, general, and complete set of

sequence features. An advantage of this framework is that the SVM can be subsequently used to scan the genome for novel enhancers not in the original training set. The results of scanning a well-studied region near *Dlx1/2* is shown in Figure 2B and detects novel and experimentally confirmed enhancers, as discussed in detail below.

[98] To evaluate classification performance, a fivefold cross validation method was used. Initially, the data set to be classified was randomly partitioned into five subsets. One subset was then reserved as a test data set, and the SVM weights were trained on sequences in the remaining four subsets. The SVM was then used to predict the reserved test data set to assess its accuracy. This process was repeated five times so that every sequence element is classified in one test set. Because there is a trade-off between specificity (the accuracy of positively classified enhancers) and sensitivity (the fraction of positive enhancers detected), the quality of the classifier was measured by calculating the area under the ROC curve (auROC), as shown for several cases in Figure 3. The five test set auROCs were averaged to give a summary statistic of the SVM performance; these five test sets generate the error bars in Figure 3.

[99] To test sensitivity to various assumptions in the SVM construction, the cross-validation experiments were repeated on each tissue-specific enhancer set using SVM classifiers with different types of kernels: spectrum kernels (Leslie et al. 2002), mismatch spectrum kernels (Leslie et al. 2004), and Gaussian kernels. The Gaussian kernel and spectrum kernel vary the functional form by which features contribute to the overall decision boundary, while the mismatch spectrum kernel retains the linear contribution of the features but uses a different set of features by allowing a certain number of base pair mismatches to a given k-mer (see Methods). In addition, a commonly used alternative approach was tested, the Naive Bayes classifier, which learns the parameters for each feature independently (the SVM learns parameters for all features at the same time). Despite this assumption of independence, the Naive Bayes classifier has performed very well on a broad range of machine learning applications.

[100] Many SVMs can successfully distinguish enhancers from random genomic sequences with auROC > 0.9, regardless of: the types of kernels, the types of tissues, or the length of the k-mers (Fig. 3; Fig. 10A). In general, larger k-mers achieved superior performance (Fig. 10A), but predictive power began to decrease when k was greater than six because of overfitting (the feature vector becomes sparse). On the other hand, Naive Bayes classifiers

were significantly less accurate in discriminating enhancers from random genomic sequences ($\text{auROC} < 0.79$), indicating that the assumption of conditional independence between k-mers in the Naive Bayes model impaired its performance. Figure 3A–C shows summaries of comparison between ROC curves of SVM (solid) and Naive Bayes (dotted). Because of its robust performance ($\text{auROC} = 0.94$) and ease of interpretation, the 6-mer spectrum kernel was chosen as the standard model for the results shown herein.

[101] Methods of the present disclosure comprise distinguishing individual enhancer sets from random genomic sequences, and distinguishing between enhancers in different tissues (forebrain, midbrain, limb). Since some enhancers are active in two or more tissues, an aspect comprises removing overlapping regions from both sets before analysis. With the full set of 6 mers, forebrain and midbrain enhancers can be discriminated from limb enhancers with a reasonable auROC of ~ 0.84 – 0.86 . However, the SVM failed to successfully discriminate forebrain and midbrain enhancers (Fig. 3D). This indicates that the compositions of TFBSs enriched in forebrain and midbrain enhancers may be similar to each other but are sufficiently different from those in limb-specific enhancers to permit classification. Significant overlap between the forebrain and midbrain enhancer sets in the original data set supported this interpretation (48.7% of midbrain enhancers are also in the forebrain set).

[102] When comparing against random genomic sequence, the size of the negative sequence set may be chosen. The genomic ratio of enhancers to nonenhancer sequence is very large (it is estimated that enhancers comprise 1%–2% of the genome in a given cell-type), and ideally alternative prediction methods would be compared using a very large negative set. However, some computational methods can not handle such large amounts of sequence due to memory constraints. To compare between data sets, the same ratio between positives and negatives was used. To test the scaling with negative set size, three negative sets (roughly balanced, $1\times$, $50\times$ larger, and $100\times$ larger than the positive enhancer set) were used. Although auROC is a standard metric, when the positive and negative sets were unbalanced, the precision-recall (P-R) curve was a more reliable measure of performance than the ROC curve. Precision was the ratio of true positives to predicted positives, and recall was identical to the true positive rate in the ROC curve. The P-R curves can be quantified by the area under the precision-recall curve (auPRC), or average precision. For the classification of EP300 forebrain (fb), limb (lb), and midbrain (mb) enhancers from genomic sequence, auROC was unaffected by the size of the negative set (Fig. 3E), but auPRC dropped (Fig. 3F) as n became large and the high-

scoring tail of the negative sequences became competitive with the true positive sequences. However, the trends of auROC and auPRC were usually consistent. Comparison of auROC and auPRC for the negative set size scaling for all positive data sets is shown in Figure 12.

Most predictive sequence elements are known transcription factor binding sites

[103] Methods of the present disclosure comprise identifying which subsets of sequence features allowed the SVM to successfully discriminate enhancers from random sequence. The SVM discriminant function was defined as the sum of weighted frequencies of k-mers in the case of the k-spectrum kernel, and the classification was determined by the sign of the discriminant function (see Methods). Therefore, k-mers with large positive and negative SVM weights indicate predictive sequence features: k mers with large positive weights are sequence features specific to enhancer sequences, and k mers with large negative weights are sequences that are present in random genomic sequence but depleted in enhancers. The SVM classification was conducted again, using only the subset of k-mers with largest positive and negative SVM weights (Fig. 10). The SVM using fifty 6-mers with the largest positive weights and another fifty 6-mers with the largest negative weights achieved auROC of 0.90 for the forebrain enhancer data set. This demonstrated that the largest weight k-mers predict enhancers with similar accuracy, although the auROC did decrease somewhat compared to the result with all k-mers (Fig. 3A–C). Interestingly, the most frequently observed k-mers did not always have the largest SVM weights or vice versa. Only a weak correlation between SVM weights and k-mer frequencies was found (Fig. 13). The most predictive single k-mer (auROC = 0.65) was AGCTGC, which was present in 60% of the true positive forebrain enhancers, but it was also present in 34% of the negative genomic regions. By combining many k-mers, the full SVM and the SVM with the 100 top k-mers achieved greater accuracy than single k-mers. The SVM's outperformance of the Naive Bayes classifier, which assumed feature independence, indicated that these features contribute cooperatively.

[104] Many of the most predictive k-mers, (those with the largest positive weights) were recognizable as binding sites for TFs known to be involved in embryonic nervous system development. Each of the predictive k-mers were scored with PWMs for known motifs available in public databases [JASPAR (Bryne et al. 2008), TRANSFAC (Matys et al. 2003), and UniPROBE (Newburger and Bulyk 2009)] using the TOMTOM package (Gupta et al. 2007). Because the databases contain many PWMs from families of TFs with similar specificity, many PWMs often score highly for a given k-mer, so for each k-mer the family of

matched TFs with q value < 0.1 was reported (Storey and Tibshirani 2003), and representative high scoring TFs within that family were listed. This mapped known TFBS to 85% of the most predictive k-mers, while only 24% of all k-mers match a known TFBS (Binomial test P -value $= 1.5 \times 10^{-8}$). Table 1A shows the fifteen 6-mers with the largest positive SVM weights. The full lists of SVM weights used in the analysis herein are provided herein. The elements that positively contribute to EP300 binding include many k-mers with TAAT or ATTA cores, which are bound by the homeodomain family (Berger et al. 2008). Several homeodomain protein genes have restricted expression in the embryonic mouse forebrain and are required for proper forebrain development, such as *Otx* and *Dlx* (Bulfone et al. 1993; Matsuo et al. 1995; Zerucha et al. 2000). Other predictive factors include the members of the basic helix-loop-helix (bHLH) family, which bind variations of E-box elements (CANNTG). Some bHLH factors are known to be crucial regulators of neural and cortical development (Lee 1997; Bertrand et al. 2002; Ross et al. 2003) and are also known to interact with the coactivator EP300/ CREBBP (Chan and La Thangue 2001).

[105] In an aspect, methods and systems of the present disclosure comprise identifying binding sites that are significantly absent or depleted in EP300 enhancers. The presence of k-mers with large negative weights in a sequence significantly decreases the likelihood that that sequence will be classified as an enhancer. Biologically, the presence of these binding sites would interfere with the operation of the enhancer in a specific tissue. ZEB1-related k-mers have the largest negative weights in forebrain enhancers (Table 1B). For example, the ZEB1 binding k-mer CAGGTA is present in 29% of the negative sequences but only 18% of the forebrain enhancer sequences. Also known as AREB6, ZEB1 (zinc finger E box binding homeobox 1) is a member of the ZEB family of transcription factors, which play crucial roles in epithelial-mesenchymal transitions (EMT) in development and in tumor metastasis by repressing transcription of several epithelial genes including E-cadherin (Vandewalle et al. 2008). Although ZEB family members can work as both activators and repressors, their depletion in EP300-bound regions implies that ZEB1 binding can disrupt EP300 activation.

[106] Although some negative weight k-mers are predictive (e.g., ZEB1), on average the positive weights in Table 1A are more predictive than the negative weights (Table 1B) for all data sets. The absolute values of most negative weight k-mers are significantly less than those of the positive weight k-mers, as shown in Figure 4 (discussed below), where each k-mer weight is plotted along the vertical axis. The asymmetry in SVM weights indicates that the

predictive features are primarily identifying k-mers that are enriched in the enhancers rather than k-mers that are enriched in random genomic sequence (or equivalently, depleted in enhancers).

Predictive sequence elements are evolutionarily conserved and positionally constrained within enhancers

[107] In their previous analysis, Visel et al. showed that most EP300-bound regions are enriched in evolutionarily constrained noncoding regions (Visel et al. 2009). However, not all sequences in the EP300-bound regions (average length 750–800 bp) are conserved; rather, several more localized peaks of conservation (10–100 bp) within the EP300-bound regions are observed in most cases. These peaks of localized conservation probably identify the smaller functional regions within a more extended enhancer. Though not wishing to be bound by any particular theory, it was thought that if the predictive k-mers reflect actual TFBSs, they would tend to be preferentially located within these evolutionarily conserved localized regions. To test this systematically, the degree to which individual k-mers were present in conserved regions was measured by averaging the phastCons conservation score (Siepel et al. 2005) over each instance of the k-mer (see Methods), and examined its correlation with SVM weight. Figure 4 shows that k-mers with large positive SVM weights are significantly more conserved than average. All but one (CCCCTC) of the 6-mers with large positive SVM weights (three or more standard deviations above the mean) have large conservation scores (at least one and a half standard deviation above the mean conservation score). While the most predictive k-mers were significantly more conserved, moderate correlation between the phastCons conservation scores and the SVM weights for all k-mers was also observed (Pearson correlation coefficient = 0.35). This evidence supports the idea that the predictive sequence features are more evolutionarily conserved than the less predictive regions within the enhancers.

[108] Since conservation was found in narrow peaks within the enhancers, it follows that there might be additional positional constraints between the predictive elements. Mechanistically, these constraints are most likely indicative of a cooperative mechanism, either involving TF-TF interactions or spatially constrained activity of individual factors. Spatial constraints between TFBSs have been observed frequently in yeast (Beer and Tavazoie 2004). In Figure 5, the distribution of minimum pairwise distances between the ten most predictive sequence elements in the forebrain enhancers (6-mers with the largest

positive weights) was compared to their distribution in the null sequences. The forebrain pairwise distance distribution was shifted to lower distances (they are closer to each other) compared to null sequences. To measure the statistical significance of this difference the pairwise distance distribution for these 6-mers in 100 different negative sets was calculated. The standard deviations of these 100 negative sets are shown as dashed lines in Figure 5, and the forebrain distribution often deviates from the null distribution by several standard deviations, especially for small spacing. The difference between the forebrain and null pairwise distance distributions can be measured by the two-sample Kolmogorov-Smirnov test, ($P\text{-value} < 2.2 \times 10^{-16}$), which further demonstrated the significant clustering of predictive sequence elements. Looking at the small spacing end of this distribution (inset in Fig. 5), periodic enrichments with characteristic spacing of 10–11 bp was observed. The highest peak was around 11 bp, almost two times higher than the null distribution. These positional correlations suggest cooperative binding interactions in phase with the 10.5 bp DNA helix periodicity, consistent with previous observations (Erives and Levine 2004; Hallikas et al. 2006), and local physical interactions between the factors that bind these DNA sequence elements.

Genome-wide SVM predictions identify novel enhancers

[109] In an aspect, methods and systems of the present disclosure comprise predicting additional functional regions that were not determined to be EP300-bound from the ChIP-seq data by scanning the entire genome systematically with the trained SVM. The mouse genome sequence was segmented into 1-kb regions with 0.5k-bp overlap, resulting in about 5.2 million overlapping sequence regions. To compare with the 2453 forebrain region “EP300 training set”, the centromeric regions, telomeric regions, and regions containing at least 70% repeats were removed, (however, this filter had minimal impact on the predictions). All of these 1-kb regions were scored using the SVM with the $k = 6$ spectrum kernel for forebrain enhancers. An example of the continuous SVM score along the *Dlx1/2* locus is shown in Figure 2B (“Raw SVM Score”). *Dlx1* and *2* are expressed in the mouse forebrain (Bulfone et al. 1993; Ghanem et al. 2003; Wigle and Eisenstat 2008). Besides the sole EP300 training set element in this region (URE2) (labeled “EP300 ChIPseq” in Fig. 1B), two other enhancers within this locus have been experimentally validated (“Known Enhancers”) (labeled i12a and i12b) (Ghanem et al. 2003). These enhancers (i12a and i12b) were detected by the trained SVM but were not in the EP300 training set because their raw sequence read density was not

above the stringent threshold used in Visel et al. (2009). Comparing the “Raw EP300ChIPseq” track to the “Raw SVM score” in Figure 2B shows correlation: Most of the predicted high scoring SVM regions have raw EP300 ChIP-seq signal significantly above background but did not have sufficient read density to be included in the EP300 training set. To support this anecdotal evidence, the genome wide correlation between these SVM predicted regions and EP300 read density was evaluated. In Figure 14, the EP300 ChIP-seq read density was plotted as a function of distance from the center of each of the top 1% SVM scoring regions. Enrichment of EP300 ChIP-seq signal around the SVM predicted regions was found, indicating that many of these predicted loci are, indeed, bound to some extent by EP300 but fall somewhat below the read threshold used to determine the EP300 training set. Figure 16 shows the correlation between SVM score and EP300 reads in all genomic 1-kb regions, showing again that there is a significant population of high scoring SVM regions enriched in EP300 signal but not in the EP300 training set.

[110] To define a high confidence set of enhancer predictions, an appropriate cutoff for the SVM score was chosen using more realistic large negative training set sizes (50× and 100× negative sequences), covering ~6%–12% of the nonrepetitive genome. The false discovery rate (the expected fraction of predicted positives which are false positives, $FP/(FP+TP)$), from the P-R curves was estimated in Figure 3F. The precision is weakly dependent on negative set size when n is large, due to the fact that the positive and negative histograms of SVM scores have a similar shape for larger negative set sizes, as shown in Figure 16. To trade off precision and recall, a cutoff that corresponds to 50% recall was chosen, which at 1× is an SVM score of 1.0. For the large negative sets, precision is ~50% when recall is 50%, and it was estimated that the false discovery rate was ~50%. In other words, at this cutoff ($SVM > 1.0$) on the training set, 50% of the EP300 training set regions and an equal number of negative regions were captured.

[111] Disclosed herein are comparisons of the properties of the SVM predicted enhancer regions ($SVM > 1.0$), the EP300 training set regions, and nonenhancer genomic regions ($SVM < 1.0$). These three sets are all distinct, i.e., each genomic 1-kb region can only belong in one class. Any 1-kb region which overlaps a training set region by as little as 1 bp is excluded from the SVM sets and included in the EP300 training set. The EP300 training set and SVM predicted regions have similar properties, much different than the nonenhancer regions.

[112] At an SVM score threshold of 1.0, 33,232 1-kb regions in the genome (outside of the EP300 training set) were predicted, or 26,920 enhancers after merging overlapping regions, and it was expected about 13,460 of these to be true enhancers. This threshold appeared to be a good tradeoff between detecting many biologically significant enhancers with an acceptable false discovery rate. The full lists of SVM scores for these regions are included as Supplementary Material. The robustness of these top SVM scoring regions was established by training separate SVMs with independent random null sequence sets as the negative class. There was extensive overlap between the top scoring regions using these different SVMs (Table 5), and the correlation of individual SVM scores between two different SVMs is high (Pearson correlation coefficient = 91.5%), as shown in Figure 17. That the SVM classifier identified many more sequence regions than the EP300 training set may be due to several factors: (1) As discussed above, these predicted regions may be false positive enhancers; (2) they may be true positive enhancers that were undetected in the ChIP experiments because of an overly stringent cutoff for defining the EP300 training set; (3) they may be true positive enhancers that are not EP300-bound in this tissue at the developmental stage of the experiment but may be EP300 bound in other tissues or times; or (4) they may be true positive enhancers that operate independently of EP300 but share some similar sequence features. All, but the first possibility, are potentially biologically interesting.

[113] In an aspect, methods and systems of the present disclosure comprise in vivo or in vitro assays or experiments to confirm the output results of test data from a trained SVM. To assess the validity of these genome-wide predictions with independent experimentation, the DNase I hypersensitivity of the high scoring forebrain SVM regions was quantified with experiments in embryonic mouse whole brain provided by the mouse ENCODE project (data available from <http://genome.ucsc.edu/ENCODE/>; J. Stamatoyannopoulos, in prep), using methods described in John et al. (2011). DNase I hypersensitivity measurements detect open or accessible chromatin, including promoters and enhancers, independent of EP300 binding. Although these DNase I experiments are not strictly specific to forebrain and were 3 d later in development, enrichment in brain hypersensitivity strongly corroborates the predictions as tissue-specific enhancers. In Figure 6, the predicted 1-kb regions from the EP300 fb trained SVM were split into four classes ($SVM < 0.5$, red; $0.5 < SVM < 1.0$, gray; $1.0 < SVM < 1.5$, cyan; and $SVM > 1.5$, blue) and one EP300 training set class (EP300-bound regions, green). The distributions of average intensity of DNase I hypersensitivity of the different SVM scoring classes were plotted in Figure 6A, which shows a dramatic increase in DNase I signal

in E14.5 brain only for high scoring SVM regions.

[114] There is no enrichment of DNase I signal for the same regions in other tissues; for example adult kidney is shown in Figure 6B as a negative control. Because the DNase I hypersensitive regions include promoters and other open regions, the converse is not true, i.e., while almost all high-scoring SVM regions have a high DNase I signal, not all high-signal DNase I regions have a high SVM score (data not shown). With this understanding, the precision and specificity with which the SVM detects DNase I sensitive enhancers was evaluated. Because the SVM score and DNase I signals are continuous, DNase I signal > 10 was considered to be positive (open chromatin), and DNase I < 2 was considered to be negative (not open) for purposes of quantification, consistent with the distributions in Figure 6A, B. Then, regions with DNase I > 10 and SVM > 1.0 are true positive predictions, and DNase I < 2 and SVM > 1.0 regions are false positive predictions. Table 2 shows the number of 1-kb genomic regions in each class. The precision is $TP/(TP+FP)$, or the accuracy of the predicted positives. The sensitivity is $1-FPR$ (false positive rate), or the fraction of negatives that were predicted to be positive. As shown in Table 2, SVM > 1.0 predictions have a 56.3% precision, and more stringent SVM > 1.5 predictions have a 74.5% precision. These results are consistent with the above estimate that 50% of these novel predictions are true enhancers functioning in mouse brain.

[115] To further support the biological significance of these novel SVM-predicted enhancers, their proximity to forebrain-expressed genes was examined. Microarray experiments (Visel et al. 2009) identified 885 (495) genes overexpressed (underexpressed) in the forebrain at E11.5. The intergenic distance between the EP300 training set regions and the transcription start site (TSS) of the nearest overexpressed genes were examined, along with the distance between the SVM- predicted enhancer regions and the overexpressed genes. All regions overlapping a training set region were omitted from the set of predictions. As shown in Figure 7, both the EP300 training set and the predicted enhancer regions are significantly enriched near (within 10 kb of) the TSS of a forebrain overexpressed gene. Notably, the SVM predicted regions with the more stringent SVM cutoff score (SVM > 2.0) are even more enriched within 10 kb of the overexpressed genes than the EP300 training set, further evidence that the SVM is capturing functional regions with spatial and temporal specificity. In comparison, randomly chosen genomic regions show no such enrichment. While the EP300 training set is not enriched near forebrain underexpressed genes, the SVM predicted

regions are significantly enriched within 10 kb of forebrain underexpressed genes (Fig. 7). Though not wishing to be bound by any particular theory, it is hypothesized that because the EP300 bound regions are not enriched near the underexpressed genes, it is unlikely that EP300 is acting as a transcriptional repressor here. It seems more likely that the SVM is predicting enhancers that are bound by EP300 in other tissues or at other times in development. These enhancers could activate the neighboring genes relative to their expression level at E11.5 in the forebrain, which would appear indistinguishable from forebrain repression. This hypothesis is supported by the fact that several of the underexpressed genes with nearby SVM- predicted enhancers play roles in nervous system development, including many HOX genes known to function in A-P axis patterning.

SVM also predicts human enhancers

[116] The present disclosure comprises use of a SVM, trained with a data set disclosed herein or the 6-mers data set disclosed herein, or a data set from a species other than humans, comprising wither homologous or nonhomologous sequences, to predict human enhancers. An aspect of the disclosure comprises use of training data comprising enhancer sequences from one species to train a SVM, wherein test data comprising sequences from a second unrelated species are used in the trained SVM to predict enhancer sequences in the second species. Such sequences used in the training data and the test data may be homologous or nonhomologous. For example, human orthologous regions (hg18) of the mouse EP300 training set with the liftOver utility from the UCSC genome browser (Karolchik et al. 2008) were found. With 70% or greater identity, 2205 of the 2453 forebrain enhancers were successfully mapped onto the human genome. 13 mapped sequences longer than 3 kb were discarded. SVMs were trained to discriminate this positive human training set from an equal number of human random sequences generated by these null model and achieved reasonably high auROC = 0.87 (Fig. 18). More stringent orthology cutoffs (requiring 90% and 95% identity instead of 70%) were tested and it was found that the overall performance was very similar (Fig. 18). Thus, an SVM trained on human sequence homologous to the mouse EP300 training set sequences is able to predict test set enhancers with only slightly reduced accuracy relative to mouse.

[117] Human enhancer regions with a SVM trained on the mouse data set was predicted, which does not require sequence alignment to identify orthologous regions. This approach is useful in situations where it is difficult or impossible to obtain similar data sets in each

species. It also provides further information about the conservation of predictive k-mers between the two species. Two raw SVM scores (one trained on the human homologous set, the other on the mouse data set) on the human genome around *Otx2* were compared, and very similar SVM score patterns were observed. Moreover, an experimentally verified enhancer (Kurokawa et al. 2004) was captured by both SVMs (Fig. 19). The entire genome was analyzed to assess how many top SVM-scoring regions overlap each other (Table 6). Although the overlaps were not as significant as scores using only different negative sets (Table 5), a large fraction of top SVM-scoring regions were still shared between the two SVMs, so to a large degree, an SVM trained on mouse can be used to successfully predict human enhancers. This result is in general agreement with *in vivo* experimental results (Wilson et al. 2008) where human DNA transplanted into mice was shown to bind mouse TFs (HNF1A, HNF4A, HNF6) in a pattern virtually indistinguishable from their binding patterns in human, indicating that variations in genomic TF binding between human and mouse are due to local DNA sequence differences, not due to evolutionary divergence of individual TF binding specificities between the two species.

Comparison between different EP300/CREBBP ChIP-seq data sets reveals sequence elements important for pluripotency

[118] Methods and systems of the present disclosure comprise *in vivo* or *in vitro* assays or experiments to confirm the output results of test data from a trained SVM. The success of the SVMs in predicting EP300 binding in mouse embryonic brain and limb motivated a comparison with other EP300/CREBBP ChIP-seq data sets. The overlap between Visel's *in vivo* data set (EP300 forebrain, midbrain, and limb) and two other data sets were examined: CREBBP-bound regions in activated cultured mouse cortical neurons (Kim et al. 2010), and EP300 bound regions in cultured mouse embryonic stem cells (Chen et al. 2008), herein referred to as "CREBBP neuron" and "EP300 ES". These data sets share similar ChIP-seq methodology, and it would address the overlap between activation mediated by the close homologs EP300 and CREBBP, and to address differences in EP300 binding in different tissues and cell populations. CREBBP neuron enhancers only overlap significantly with EP300 forebrain enhancers (not midbrain or limb) (Table 7). EP300 ES enhancers do not significantly overlap with any other set (fb, mb, lb, or CREBBP neuron) (Table 7). This indicates that EP300-mediated embryonic neuronal development is linked to CREBBP-mediated neural activity dependent transcription via extensively shared common regulatory

regions. It was observed that several predictive k-mers with large positive weights, such as homeodomain binding sites (TAAT core) and bHLH domain binding sites (E box, CANNTG), were shared between the two data sets (Table 1A; Table 8), which further indicated common modes of regulation.

[119] Figure 3G shows ROC curves discriminating CREBBP neurons (auROC = 0.93) and EP300 ES (auROC = 0.77) from random genomic sequences. The lower EP300 ES auROC is partly due to the relatively smaller number of regions bound in the EP300 ES positive set. Also, the EP300 ES data set contains a larger fraction of repeat sequences, indicating that this data set may be less specific for functional EP300 binding. Nonetheless, SVMs still can extract informative k-mers from this data set and can largely discriminate the EP300 ES set from random genomic sequences. Alternatively, instead of comparing to random genomic sequence, these sets (EP300 forebrain, CREBBP neuron, EP300 ES) were successfully classified against each other, as shown in Figure 3H. It is interesting to note that EP300 forebrain can be discriminated from CREBBP neuron with high auROC, even though they share many regions and have some common predictive k-mers (homeodomain, SOX, bHLH) when classified against random sequence (Table 1A; Table 8). However, when classified against each other, it was observed that the predictive k-mers specific for EP300 forebrain remain homeodomain, SOX, and bHLH, but the k-mers predictive for CREBBP neurons become nuclear factor I (NFI), activator protein 1 (AP1), and cyclic AMP-responsive element-binding protein (CREB) binding sites (Table 10). Therefore, homeodomain, SOX, and bHLH binding sites may play more prominent roles in neural developmental processes than in neural activity dependent transcription.

[120] The biological significance of the predictive k-mers in these new data sets was assessed. Most of the predictive k-mers can be related to known TFBSs (Tables 8, 9), and that many of the identified TFBSs were involved in signaling pathways known to function in the relevant experimental conditions. For the CREBBP neuron data set, AP1 related 6-mers, GACTCA and TGACTC, the first and third largest weights respectively (Table 8), were the target of heterodimers of the regulators Fos and Jun, which play critical roles in neural activity dependent transcription regulation (Flavell and Greenberg 2008). CREB, which directly interacts with CREBBP, was also essential for the activation of several genes in response to neural stimulation, and its binding site is ranked fourth in Table 8 (Flavell and Greenberg 2008; Kim et al. 2010). Kim et al. noted that two other transcription factors,

neuronal PAS domain containing protein 4 (NPAS4) and serum response factor (SRF) as well as CREB, strongly colocalize with CREBBP binding regions. NPAS4 contains a bHLH domain, and its canonical binding sites, E-box elements, are ranked at second and sixth in Table 8. The SRF binding site is also known as a CArG box, whose consensus sequence is CCWTATAWGG (SEQ ID NO:1) (Bryne et al. 2008). A specific k-mer instance of the CArG box is ATATGG, ranked at 17th with $w = 3.00$, just below the top fifteen in Table 8. Therefore, all well-characterized TFBSs known to play a role in neuronal activation were successfully captured by this SVM. Two additional transcription factor families also scored highly in the CREBBP neuron data set: homeodomain and NFI. These families have been discussed little in this context, although it is known that both NFI and homeodomain transcription factors are key regulators of central nervous system development (Wilson and Koopman 2002; Mason et al. 2008). One relevant example of neural activity-dependent expression of a homeobox protein was found, LMX1B (Demarque and Spitzer 2010). There may be still unknown mechanisms involving NFI and homeodomain proteins in the context of neural activity-dependent transcriptional regulation, but broadly speaking, the results indicate significant pleiotropy between neuronal developmental pathways and neural activity-dependent signaling pathways.

[121] Comparison of the EP300 ES data to CREBBP neuron and EP300 forebrain can address which binding sites and factors are responsible for maintaining a differentiated or pluripotent state. For the EP300 ES data set, a method disclosed herein identified factors known to be crucial for maintaining ES identity: high scoring binding sites for NANOG-POU5F1(also known as OCT4)-SOX2 SOX-family factors (Table 9) were found, essentially the same binding sites found in previous studies (Pavesi et al. 2001; Chen et al. 2008). A uniform approach was used to map k-mers to TFBS in the databases, but there is substantial overlap in many TF specificities, and some reported matrices may score higher than the biologically relevant database entry. For instance, in Table 9, the high-scoring matrices (SOX17, POU2F1, and POU3F3) appear on the list instead of the relevant SOX2, POU5F1, and NANOG, which have nearly identical binding sites. SOX2, POU5F1, and NANOG bind a combination of the SOX2 (CATTGT) and POU5F1 (ATGCAAAT) consensus sites (Chen et al. 2008), and the 6-mer subsequences within the combined binding site (CATTGTYATGCAAAT (SEQ ID NO:2)) have high SVM weights. Table 3 shows how large weight k-mers tile across this extended known binding site. Positive weight binding sites for ESRRB and STAT3 were found, which are known to be frequently located nearby

the NANOG-POU5F1-SOX2 clusters assessed by ChIP-seq analysis (Chen et al. 2008). Many of the positive weight EP300 ES k-mers (ESRRB, RORA1/2, PPARG) are among the largest negative weights in CREBBP neuron (Table 9), indicating that binding sites for factors responsible for maintaining pluripotency are significantly absent from neuronal enhancers (CREBBP neuron), as would be expected given the developmental maturity of neurons.

SVM can predict other ChIP-seq data sets

[122] The present disclosure comprises SVM methods to classify and detect EP300/CREBBP- bound enhancers, or any data set which may be framed as a sequence classification: e.g., ChIP- seq, ChIP-chip, or DNase I hypersensitivity data sets. In these situations, the SVM can be used to identify primary binding sites in regions identified by transcription factor ChIP experiments and may also identify binding sites for secondary factors colocalized with the ChIPed TF or binding sites significantly depleted in the functionally occupied regions. Current de novo motif- finding methods such as AlignACE (Hughes et al. 2000) or MEME (Bailey and Elkan 1994) have limited success when applied to data sets of this size. When run on the forebrain enhancer data set, AlignACE (when it converged) failed to report any meaningful motifs. While Chen et al. (2008) did successfully identify SOX2, POU5F1 (OCT4), and NANOG binding sites in the EP300 ES data with Weeder (Pavesi et al. 2001), the EP300 ES data set was the smallest and least diverse of the data sets analyzed.

[123] To directly assess the ability of a trained SVM to predict binding of individual transcription factors, ChIP-seq results on the TF ZNF263 were analyzed. ZNF263, a 9-finger C2H2 zinc finger which is predicted to have a binding site of ~24 bp, was chosen to assess how well k-mers can represent extended degenerate binding sites. ChIP-seq data on ZNF263 in a K562b cell line (Fietze et al. 2010) was used which identified 1418 strongly bound regions. Predicting against a 503 random negative set yielded auROC = 0.938 and auPRC = 0.51 (Fig. 21B, D). Many of the largest weight k-mers are subsequences within the large PWM found by de novo motif-finding tools applied to this data set (Fietze et al. 2010), and the SVM is combining k-mers which tile across the binding site to achieve high predictive accuracy. The k-mer GAGCAC also received a large weight. This indicates that the present disclosure should have significant predictive value for a wide range of binding data.

Comparison to alternative approaches

[124] As an alternative to k-mers, known PWMs were used as features in an SVM. 811 PWMs from existing databases of known TF specificities [JASPAR (Bryne et al. 2008), TRANSFAC (Matys et al. 2003), and UniPROBE (Newburger and Bulyk 2009)] were used. When using these features, the highest PWM scores in each sequence for each matrix was used as the feature vector. This 811-PWM SVM was able to achieve $\text{auROC} = 0.87$ for forebrain enhancers (compared to $\text{auROC} = 0.93$ for k-mers), somewhat less predictive than the k-mer approach (Fig. 21A), against a 503 random negative set. However, this translates into a significantly lower $\text{auPRC} = 0.22$ (compared to $\text{auPRC} = 0.43$ for k-mers) (Fig. 21B). The optimal combined weighting of the known PWMs and 6-mers features (2080 + 811 total features) gives marginal improvement ($\text{auROC} = 0.93$ and $\text{auPRC} = 0.49$) over 6-mers alone. The 811-PWM SVM was applied to the ZNF263 data set, which achieved $\text{auROC} = 0.83$ (compared to $\text{auROC} = 0.94$ for k-mers), reflecting the fact that accurate PWMs for ZNF263 were absent from the databases (Fig. 21B,D). The seemingly small change in auROC corresponded to a large drop in $\text{auPRC} = 0.14$, compared to $\text{auPRC} = 0.51$ for k-mers. This demonstrates that using sequence features from an unbiased and complete set can be more valuable than using an incomplete set of more accurate features (PWMs). Using the set of known TF PWMs is less predictive than the k-mer SVM, but a more complete set of PWMs might perform better. Combining the predictive k-mers into a more general PWM via a method similar to positional oligomer importance matrices (POIMs) (Sonnenburg et al. 2008) might allow clearer identification of informative sequence features from within the k-mer SVM but would not affect predictive performance.

[125] Alternative kernel methods were compared. The weighted degree kernel with shifts (WDS) (Rätsch et al. 2005) was applied to the CREBBP neuron data set (as WDS requires input sequences of equal length) and found $\text{auROC} = 0.83$, compared to $\text{auROC} = 0.93$ for the k-mer trained SVM. A notable SVM based approach which incorporates positional information between general k-mer features (KIRMES) has been recently described (Schultheiss et al. 2009; Schultheiss 2010). This package was applied to the forebrain EP300 data set and found $\text{auROC} = 0.90$. In the current implementation of KIRMES, k-mers are selected by their relative frequency in the positive set, and it is likely that further optimization would make this approach comparable to the k-mer SVM result. Additionally, the periodic spatial distribution in Figure 5 suggests that a model based on difference in angle (similar to Hallikas

et al. 2006) would be more appropriate than the Gaussian spatial dependence used in KIRMES. Another approach to predict promoters (Megraw et al. 2009) used PWMs and 11-logistic regression. Little difference was found between logistic regression and SVM: Using the k-mer feature vectors in 11-logistic regression yielded $\text{auROC} = 0.92$ on the EP300 forebrain data set, using publicly available software (Koh et al. 2007).

[126] Disclosed herein are methods and systems comprising a support vector machine to accurately identify regulatory sequences without any prior knowledge about transcription factor binding sites, using only general genomic sequence information. While the ROC and P-R curves demonstrate that the trained k-mer SVM was able to identify enhancers based on their sequence features, the biological relevance of the predicted enhancers is further supported by the following: (1) Most of the predictive sequence features identified by these methods are binding sites of previously characterized TFBSs known to play a role in the relevant context; (2) the enriched predictive sequence features are much more evolutionarily conserved within the enhancers than the less predictive sequence features, which suggests that the predictive features are under selection and comprise the functional subset of the larger enhancer regions; (3) these sequence features are significantly more spatially clustered in the enhancers than would be expected by chance, also a well-known characteristic of functional binding sites; (4) genomic regions with high forebrain SVM scores are strongly enriched in DNase I hypersensitivity signals in mouse brain but not in other tissues; (5) the predicted enhancers frequently overlap with regions of enhanced ChIP-seq signals but are somewhat below the signal cutoff necessary to be included in the original EP300 training set; and (6) these novel predicted enhancers are preferentially positioned near biologically relevant genes, and many have been experimentally verified in other studies, which further supports their biological relevance and functional roles.

[127] When scanning the whole genome to predict putative enhancers, it was predicted that 50% of the 26,920 nonoverlapping enhancers with forebrain SVM scores above 1.0 are true positives. This is a conservative estimate of the ability of the methods and systems disclosed herein to detect novel enhancers, since, when scanning the genome, 1-kb arbitrarily delimited chunks of sequence were scored; more accurate predictions might be possible by varying the endpoints of the predicted regions. Nevertheless, this genome-wide scan discovers thousands of novel predicted enhancers that were not in the original experimental training set. Methods and systems of the present disclosure can predict human enhancers based on these mouse

enhancer experiments by measuring the overlap between human enhancers predicted by an SVM trained on the mouse sequence and comparing these predictions to an SVM trained on human sequence orthologous to the mouse enhancer sequences. Finally, by comparing between other EP300/CREBBP ChIP-seq data sets, sequence features that are able to differentiate between enhancers that operate in different tissues or at different developmental stages were found. Some of these sequence features are enriched in enhancers in one specific tissue or state, but other predictive elements are notably depleted in some classes of enhancers.

[128] It is perhaps surprising that such a simple description of sequence features (k-mer frequencies) is able to classify enhancers and ChIP-seq data so well. The SVM is apparently combining k-mer features in a sufficiently flexible way to reflect combinations of binding sites and/or sequence signals which modulate chromatin accessibility. Developing an optimal sequence feature vector remains an area for future work; however, these results showing that the SVM is more accurate than Naive Bayes suggests that successful prediction requires the ability to combine features without evaluating them independently.

[129] Improvements to the methods and systems described herein, to make more accurate predictions, are theorized. Though not wishing to be bound by any particular theory, incorporating positional constraints between the features may improve the accuracy of the predictions, consistent with the observation of nonrandom spatial distributions between predictive features in the SVM. Kernel approaches have been developed which incorporate positional information, but most have been developed in the context of positional constraints relative to a single preferred genomic location or anchor point. In application to other problems, positional information relative to a transcription start site (Sonnenburg et al. 2006b), to a splice site (Rätsch et al. 2005; Sonnenburg et al. 2007), or to a translational start site (Meinicke et al. 2004) has been implemented in SVM contexts. Positional preference relative to a mean anchor point has been incorporated in a de novo motif discovery method developed by Keilwagen et al. (2011). However, the aforementioned methods are not strictly appropriate to the biological problem of enhancer detection, because enhancers have no such preferred fixed location, and the relevant positional constraints are between sequence features within the enhancer. Many approaches have modeled clusters of known binding sites (for review, see Su et al. 2010) but have limited application to mammalian enhancer prediction.

[130] Although evidence is provided that k-mer trained SVM-predicted regions are likely functional, the predicted enhancers may be based on sequence features which are tissue-specific. Alternatively, sequence features could be general to larger classes of enhancers. These common features could allow access, could stabilize, or could be recognized by generic components of the enhanceosome (Thanos and Maniatis 1995; Maniatis et al. 1998), whose activity could be modulated by tissue-specific factors, much as Pol II operates generally. Ultimately this should be determined by individual experiments. The methods and systems disclosed herein determined enhancers computationally by investigating overlaps between forebrain and limb-specific predicted regions, which were compared with the overlaps between EP300-enriched regions in forebrain and limb. For this comparison, the EP300-enriched regions were determined from the raw data set using the same threshold criteria as the previous study (Visel et al. 2009) except that fixed-length 1-kb regions were used, rather than the ChIP-seq determined peak regions. With a 1% false discovery rate (FDR), 3390 EP300-enriched regions of forebrain and 2607 regions of limb were found. Visel's EP300-bound regions are highly tissue-specific; there are only 243 regions (7%–9%) shared by the two sets. For the SVM predictions, a significantly larger fraction of forebrain predicted regions (6104 out of 39,714, 15%) were found in 34% of the limb predicted regions (18,027). This suggests that the SVMs learn features that are generally enriched in enhancers, in addition to tissue-specific sequence features. As a result, two SVMs trained on entirely different data sets can predict common regions that have general enhancer function. Moreover, the 6104 regions predicted by both limb and forebrain SVMs overlap with small EP300 peaks that are somewhat below the conservative threshold ($\text{FDR} < 0.01$); almost 50% have peak in at least one tissue. This observation further supports an hypothesis that SVM-predicted regions are likely to be functional. A further complication is that individual tissues consist of heterogeneous populations of cell types, and enhancers predicted in distinct tissues may only be active in subsets of cell types.

METHODS of USING THE SVM

[131] Once determinative sequences are found by the learning machines of the present disclosure, methods and compositions for treatments of organisms can be employed. For example, therapeutic agents can be administered to antagonize or agonize, enhance or inhibit activities, presence, or synthesis of the gene products. Therapeutic agents include, but are not limited to, gene therapies such as sense or antisense polynucleotides, DNA or RNA analogs,

pharmaceutical agents, biological molecules, small molecules, and derivatives, analogs and metabolic products of such agents.

[132] Methods and systems disclosed herein are used for identifying functional regulatory variants.

[133] Most variants implicated in common human disease by Genome-Wide Association Studies (GWAS) lie in non-coding sequence intervals. Despite the suggestion that regulatory element disruption represents a common theme, identifying causal risk variants within indicted genomic regions remains a significant challenge. The present disclosure provides sequence-based computational methods to predict the effect of regulatory variation, using a classifier (gkm-SVM)² which encodes cell-specific regulatory sequence vocabularies. The induced change in the gkm-SVM score, deltaSVM, quantifies the effect of variants. deltaSVM accurately predicts the impact of SNPs on DNase I sensitivity in their native genomic context, and accurately predicts the results of dense mutagenesis of several enhancers in reporter assays. Previously validated GWAS SNPs yield large deltaSVM scores, and novel risk SNPs are disclosed for several autoimmune diseases and other pathologies. Methods and systems disclosed herein comprising a deltaSVM provides a powerful computational approach for systematically identifying functional regulatory variants.

[134] Though not wishing to be bound by any particular theory, sequence variation in DNA regulatory elements is thought to contribute substantially to risk for common diseases. Variants associated with human disease by GWAS occur within putative regulatory elements far more often than expected by chance, suggesting that disruption of regulatory function is a common mechanism by which non-coding sequence variants contribute to human disease. Linkage disequilibrium (LD), and the absence of regulatory vocabularies, complicates the discrimination of regulatory risk variants from other variation within disease-associated intervals. The present disclosure provides methods to predict the impact of regulatory sequence variation, expediting targeted functional validation and the exploration of disease-implicated pathways. The present disclosure provides computational methods and systems to predict the impact of Single Nucleotide Polymorphisms (SNPs) on regulatory element activity.

[135] Regulatory elements modulate the expression of their target genes through direct binding of sequence-specific transcription factors (TFs). While consensus on the mechanisms of regulatory element activity is emerging, it is currently thought that what is lacking is a predictive model capable of (1) specifying the cell types and environmental conditions under which an element would modulate the expression of its target gene(s), and (2) describing how specific mutations to that sequence would influence its activity. The present disclosure provides methods and systems that addresses the latter: given a regulatory element active in a specific cell type, compute the effect of a given DNA sequence variation within the element. When trained on a set of putative regulatory sequences, the disclosed gapped k-mer Support Vector Machine (gkm-SVM)² identifies sequence features within these regulatory regions which determine their cell-type dependent activity. The disclosed gkm-SVM is used to quantify the effect of sequence changes within regulatory elements via a metric termed deltaSVM. This systematic, quantitative method and systems may comprise high quality catalogs of human regulatory elements, generated using DNase I Hypersensitivity, distinctive histone modifications, and TF binding. For example, if the gkm-SVM is trained on DNase I Hypersensitive Sites (DHSs), it identifies the sequence features that determine chromatin accessibility in the corresponding cellular context. The method optionally does not consider extant databases or binding motif data, and consequently the methods and systems can uncover novel motifs, combinatorial constraints, and key accessory factors, and quantify the significance of their individual contributions to regulatory element activity.

[136] Disclosed herein are methods and systems for a properly trained SVM which can predict cell-type specific regulatory elements from primary genome sequence alone. Such a SVM-based approach was adapted to predict the functional consequence of sequence variation within regulatory elements. This can be accomplished with a large set of dsQTLs (DNase I Sensitivity Quantitative Trait Loci) identified in a collection of human Lymphoblastoid Cell Lines (LCLs). These are SNPs within putative regulatory regions (marked by DNase I Hypersensitivity) and are associated with altered DNase I sensitivity therein. First, a gkm-SVM was trained on the top DHSs in the LCL GM128788. The gkm-SVM produced a scoring function characterized by a set of weights quantifying the contribution each possible 10-mer to a region's DNase I sensitivity in GM12878 cells. The deltaSVM was calculated, wherein the deltaSVM is the predicted impact of any Single Nucleotide Variant (SNV) on chromatin accessibility in LCLs, by summing the change in weight between alleles for each of the ten 10-mers encompassing the SNV, as shown in Fig.

43A for the dsQTL rs495322313. In 43, the indicted SNP allele disrupts a NF- κ B binding site (43b), which reduces the strong positive contribution of several 10-mers. Two neighboring SNPs do not make significant changes to the weights, as shown graphically in Fig. 43b, and the score of each allele is the sum of the weights across this region (See Figure 44). Similarly, this method can be extended to INDELs and multiple substitutions by summing weights across all affected bases. In an aspect, the present disclosure provides methods comprising INDELs (insertions or deletions in the DNA). In an aspect, the present disclosure provides methods comprising multiple substitutions.

[137] To assess the ability of deltaSVM to predict the impact of SNPs on DNase I sensitivity, deltaSVM was compared to the set of dsQTLs13, quantified by the effect-size, beta. The correlation between deltaSVM and effect-size for the 579 SNPs within 100 bp of a DHS was highly significant, Pearson correlation coefficient $C=0.721$ (t-distribution $P=7.68e-94$) (Fig. 1c). This correlation fell off rapidly with distance (See Fig. 44), thus the analysis is consistent with local action of dsQTLs. However, if the predictions are accurate, it would be found that deltaSVM analyses on non-dsQTL SNPs yield low scores, limiting false positive predictions. A 50x larger negative set of non-dsQTL SNPs with comparable levels of DNase I sensitivity was used as a negative set, since there are typically 50-100 SNPs within a single LD block. In Fig. 43d it is shown that the Receiver Operating Characteristic (ROC) curve, plotting True Positive rate (TP/P) vs. False Positive rate (FP/P), and in Fig. 43e it is shown that the Precision-Recall (PR) curve, plotting precision (true positives over predicted positives, $TP/PP=TP/(TP+FP)$) vs. recall (TP/P), for the disclosed method (gkm-SVM deltaSVM) compared favorable to four other methods.

[138] As is generally the situation for genomic predictions where the search space is large, the lower left corner of the ROC curve, where the FP rate is low, has the most dramatic effect on the accuracy (precision) of the predictions. At a recall of 10%, the gkm-SVM predictions are 55.9% accurate, ~5x more accurate than deltaSVM based on smaller 6-mers (kmer-SVM) as shown in Fig. 43e, because while the kmer-SVM can predict full regions very accurately by averaging many weights, the kmer weights needed to evaluate SNPs are determined from a small set of support vectors and are noisy. By contrast, the gkm-SVM reduces the false positive rate significantly by using much more statistically robust gapped-kmer weights. Additionally, in comparison to conservation (GERP score), and to two recently published methods integrating functional genomic datasets to predict the deleteriousness of noncoding

variants (CADD5 and GWAVA6) the gkm-SVM is $\geq 10\times$ more accurate than any of these existing methods at 10% recall (Fig. 43e). Two features contribute to the improved accuracy. First, gkm-SVM was trained on a large set (thousands) of both positive and negative elements in the relevant cell type to statistically determine the DNA sequence elements required for activity, rather than relying on the precise state of any specific regulatory element in a specific assay. Second, identify a complete catalog of both positive and negative sequence features, as many SNPs result in a significant deltaSVM based on what the variant changes to, rather than what it was in the reference/assayed genome. In the disclosed discriminative approach, gkm-SVM identified these negative sequence elements by their presence in the negative set and their absence in the positive set. This may be needed for accurately assessing the effect of variants.

[139] Methods and systems herein are used to determine how a variant modulates the expression of its target genes. 125 of 579 dsQTLs are also eQTLs¹⁹ (variants associated with differential gene expression), but some dsQTLs are anti-correlated with eQTLs¹³. Both classes of dsQTLs were strongly positively correlated with deltaSVM (Figure 46). Thus surprisingly, but consistent with earlier analysis, as 22% of the dsQTLs become more accessible, they repress target gene expression. The relationship between deltaSVM and evolutionary sequence conservation was analyzed. Interestingly, although bases predicted to either reduce or increase DNase I sensitivity when mutated were more conserved than bases predicted to be neutral, negative deltaSVM bases were much more conserved than positive deltaSVM bases (Figure 47).

[140] To directly test the ability of disclosed SVM-based methods to predict the functional consequence of sequence variation on enhancer activity, well-characterized enhancers of the pigmentation genes Tyr and Tyrp (Fig. 48 a,b) were used. A melanocyte-specific gkm-SVM was trained on a large set of putative melanocyte enhancers marked by EP300 and H3K4me112, and scored all possible SNVs in the Tyr and Tyrp1 enhancers, selecting and synthesizing more than 40 SNVs, across a range of deltaSVM scores, and tested each variant independently in luciferase reporter assays. For both enhancers, deltaSVM was strongly correlated with the observed difference in luciferase reporter activity between mutant and wild-type enhancer constructs (Pearson $C=0.778$, $P<2e-5$ for Tyr, and $C=0.529$, $P<.0095$ for Tyrp1; Fig. 48 c, d).

[141] Despite their depth, the analyses of the Tyr and Tyrp1 enhancers tested only a subset of all possible variants therein, and relied on in vitro reporter assays. A dataset in which all possible variants within a 259 bp liver-specific enhancer of the ALDOB gene were tested using a massively parallel reporter assay in vivo in mouse liver. A gkm-SVM was trained on a large set of putative liver enhancers marked by DNase I hypersensitivity and H3K4me1 signal in adult mouse liver. deltaSVM was then compared for each of the tested mutant regions to the observed functional output. There was a very high correlation ($C=0.630$, $P<3.24e-81$) between the predicted impact of the mutation using the sequence-based model and the observed change in enhancer activity relative to wild-type sequence (Fig. 49a). If the “aggregate score” model was used, averaging deltaSVM for each of the 3 possible base substitutions, this correlation reaches $C=0.691$ (Figure 50). deltaSVM was then queried for performance in predicting functional variants in diverse sets of enhancers. Data was used from another massively parallel reporter assay using targeted mutation of enhancers predicted to be active in K562 and HepG2 cells. For each wild-type construct that was expressed significantly in either cell line, all 1 bp and motif scrambling mutations were scored using a gkm-SVM trained on K562 and HepG2 DHS regions, and compared the measured expression change to the predicted deltaSVM score in each cell line. For both datasets there was a high correlation ($C=0.626$, $P<1.34e-31$ for K562 and $C=0.646$, $P<3.84e-34$ for HepG2). Since all elements were tested and scored in both cell types, this high correlation underscores the accuracy of deltaSVM’s cell-type specific predictions and is further supported by the low correlation of deltaSVMs scores from gkm-SVMs trained on non-relevant cell-types (Table 13).

Table 13. Cell-type specificity of deltaSVM predictions.

Gkm-SVM	LCL-dsQTL	Tyr	Tyrp1	Aldob	K562 enhancers	HepG2 enhancers
GM12878 DHS	0.721 (7.68e-94)	0.302 (0.172)	0.117 (0.595)	0.112 (0.00256)	0.204 (0.00062)	0.201 (0.00076)
Mouse melan-a EP300	0.245 (2.19e-9)	0.78 (2.0e-5)	0.53 (0.0095)	0.147 (7.42e-5)	0.204 (0.00062)	0.194 (0.00116)
Mouse liver DHS	0.131 (0.00157)	0.282 (0.203)	0.056 (0.798)	0.630 (3.24e-81)	-0.329 (2.04e-8)	0.551 (2.07e-23)
K562 DHS	0.581 (1.45e-53)	0.390 (0.0726)	0.104 (0.638)	0.092 (0.0137)	0.626 (1.34e-31)	-0.042 (0.483)
HepG2 DHS	0.518 (3.84e-41)	0.551 (0.00791)	0.166 (0.450)	0.547 (1.01e-57)	-0.184 (.0021)	0.646 (3.84e-34)

Weights trained on all five cell types were compared to the measured expression changes in all five cell lines. P-values are in parenthesis. Although some TFs are active in more than one cell type, generally, deltaSVM predictions are cell-type specific.

[142] In an aspect, a deltaSVM can predict the functional consequences of studied disease-associated sequence variants. deltaSVM values were compared for three experimentally validated SNPs, each of which has been shown to alter expression leading to increased disease risk or pertinent traits: Rfx6 (rs339331, prostate cancer), Bcl11a (rs1427407, fetal hemoglobin levels), and Sort1 (rs12740374, LDL cholesterol levels). Three separate gkm-SVMs were trained with DHSs from cell lines appropriate to each phenotype (LNCaP, mouse MEL, and HepG2 hepatocytes). In each case (Fig 51a-c), the expression perturbing SNP scores higher than flanking SNPs. Since the set of these validated regulatory SNPs was limited, 413 SNPs that are associated with 11 autoimmune diseases enriched in T helper cell type 1 (Th1) H3K27Ac regions were studied. A gkm-SVM was trained on Th1 DHSs8, and for each disease associated locus, scored the lead SNPs and an additional 2700 SNPs in tight LD (as defined in methods), and random SNPs including equivalent size flanking sets as a control. An example locus in BACH2, associated with several autoimmune diseases, is shown in Fig. 51d. High scoring deltaSVM SNPs were identified for 17 independent disease associations, which were predicted to be expression perturbing SNPs with high confidence ($P < .02$), while at this threshold random sampling produced 8 SNPs (Fig. 51e, f, Figure 52). Most of these high scoring SNPs were not the lead SNP, and thus represent novel predictions for the causal SNP. To make very high confidence predictions, these studies focused on the highest deltaSVM scores, but comparison with validated SNPs (Fig. 51a-c) showed that many more moderate deltaSVM scoring SNPs will also perturb regulatory activity, but with weaker phenotypic effect, depending on the function of the target gene and on variation elsewhere in the genome. In this sense the random control sampling used was highly conservative, as the positive loci are all known to be associated with disease. The high accuracy and low false positive rate of deltaSVMs (Fig. 43e) identify these causal SNPs with high accuracy. Together with the data from the lymphoblastoid dsQTLs, the Tyr and Tyrp1 enhancers, the ALDOB enhancer analyses and those performed in K562, and HepG2, these results clearly demonstrated that deltaSVM can broadly predict the empirically measured, cell-type specific functional consequences of enhancer sequence variants.

[143] The positive predictive value of deltaSVM is based on training gkm-SVM on a set of active regions to identify the cell-type specific regulatory vocabulary. Precise variant evaluation requires an accurate assessment of the relative contribution of moderate and weak binding sites or other variants which affect chromatin accessibility, which is estimated to require over 2000 training elements and a robust classifier. Table 1 shows that deltaSVM

predictions are cell type specific, i.e. deltaSVM from weights trained on one cell type are weak predictors of expression changes in other cell types. Similarly, deltaSVM only identifies the validated disease associated SNPs shown in Fig 51a-c if trained on an appropriate cell type. While the ENCODE and Roadmap projects have provided a wealth of such training data, these methods and systems comprise coupling sequence-based computational analysis with the generation of functional genomics data targeting disease relevant developmental stages and cell types.

Diagnosing diseases or pathologies Autoimmune

[144] Using a trained SVM to determine the delta SVM for identifying predictive variant sequences in the the genome of a subject leads to the use of the identified predictive variant sequences for diagnosis of the subject as having the predicted disease or pathology. Though the present disclosure shows that predictive variant sequences can be determined for autoimmune diseases, the present disclosure is not limited to just autoimmune diseases, but the methods and systems herein can be used to determine predictive variant sequences for any disease or pathology due to an alteration in the DNA or RNA sequence of a subject, such as a SNP, insertion or deletion (INDEL). Such an alteration in the DNA or RNA sequence of a subject is seen in diseases and pathologies such as cancer, congenital genetic mutation diseases, Fragile X, Down's Syndrome, cystic fibrosis, Marfan syndrome, Huntington's disease, hemochromatosis, and others known to those of skill in the art. For example, for autoimmune diseases, the present methods and systems identified predictive variant sequences for the autoimmune disease is Type 1 Diabetes, Crohn's Disease, Multiple Sclerosis, Celiac Disease, Primary Biliary Cirrhosis, Rheumatoid Arthritis, Allergy, Autoimmune Thyroid Disease, Ulcerative Colitis, Vitiligo, and Systemic Lupus Erythematosus.

[145] This disclosure is further illustrated by the following examples, which are not to be construed in any way as imposing limitations upon the scope thereof. On the contrary, it is to be clearly understood that resort may be had to various other embodiments, modifications, and equivalents thereof which, after reading the description herein, may suggest themselves to those skilled in the art without departing from the spirit of the present disclosure and/or the scope of the appended claims.

[146] All patents, patent applications, and references cited are herein expressly incorporated by reference.

[147] As used in the specification and the appended claims, the singular forms “a,” “an” and “the” include plural referents unless the context clearly dictates otherwise. Thus, for example, reference to “a pharmaceutical carrier” includes mixtures of two or more such carriers, and the like.

[148] Ranges can be expressed herein as from “about” one particular value, and/or to “about” another particular value. When such a range is expressed, another embodiment includes from the one particular value and/or to the other particular value. Similarly, when values are expressed as approximations, by use of the antecedent “about,” it will be understood that the particular value forms another embodiment. It will be further understood that the endpoints of each of the ranges are significant both in relation to the other endpoint, and independently of the other endpoint. It is also understood that there are a number of values disclosed herein, and that each value is also herein disclosed as “about” that particular value in addition to the value itself. For example, if the value “10” is disclosed, then “about 10” is also disclosed. It is also understood that when a value is disclosed that “less than or equal to” the value, “greater than or equal to the value” and possible ranges between values are also disclosed, as appropriately understood by the skilled artisan. For example, if the value “10” is disclosed the “less than or equal to 10” as well as “greater than or equal to 10” is also disclosed. It is also understood that the throughout the application, data is provided in a number of different formats, and that this data, represents endpoints and starting points, and ranges for any combination of the data points. For example, if a particular data point “10” and a particular data point 15 are disclosed, it is understood that greater than, greater than or equal to, less than, less than or equal to, and equal to 10 and 15 are considered disclosed as well as between 10 and 15. It is also understood that each unit between two particular units are also disclosed. For example, if 10 and 15 are disclosed, then 11, 12, 13, and 14 are also disclosed.

[149] In this specification and in the claims which follow, reference will be made to a number of terms which shall be defined to have the following meanings:

[150] “Optional” or “optionally” means that the subsequently described event or circumstance may or may not occur, and that the description includes instances where said event or circumstance occurs and instances where it does not.

[151] “Probes” are molecules capable of interacting with a target nucleic acid, typically in a sequence specific manner, for example through hybridization. The hybridization of nucleic acids is well understood in the art and discussed herein. Typically a probe can be made from any combination of nucleotides or nucleotide derivatives or analogs available in the art.

[152] As used herein, the term “subject” refers to the target of administration, e.g., an animal. The term “subject” also includes domesticated animals (e.g., cats, dogs, etc.), livestock (e.g., cattle, horses, pigs, sheep, goats, etc.), and laboratory animals (e.g., mouse, rabbit, rat, guinea pig, fruit fly, etc.). Thus, the subject of the herein disclosed methods can be a vertebrate, such as a mammal, a fish, a bird, a reptile, or an amphibian. Alternatively, the subject of the herein disclosed methods can be a human, non-human primate, horse, pig, rabbit, dog, sheep, goat, cow, cat, guinea pig, or rodent. The term does not denote a particular age or sex. Thus, adult and newborn subjects, as well as fetuses, whether male or female, are intended to be covered. In an aspect, a subject can be a human patient.

[153] As used herein, the term “treatment” or “treating” refers to the medical management of a patient with the intent to cure, ameliorate, stabilize, or prevent a disease, pathological condition, or disorder (such as, for example, a skin disease or disorder, an inflammatory disease or disorder, or a heart disease or disorder (i.e., a myocardial infarction)). This term includes active treatment, that is, treatment directed specifically toward the improvement of a disease, pathological condition, or disorder, and also includes causal treatment, that is, treatment directed toward removal of the cause of the associated disease, pathological condition, or disorder. In addition, this term includes palliative treatment, that is, treatment designed for the relief of symptoms rather than the curing of the disease, pathological condition, or disorder; preventative treatment, that is, treatment directed to minimizing or partially or completely inhibiting the development of the associated disease, pathological condition, or disorder; and supportive treatment, that is, treatment employed to supplement another specific therapy directed toward the improvement of the associated disease, pathological condition, or disorder. In various aspects, the term covers any treatment of a subject, including a mammal (e.g., a human), and includes: (i) preventing the disease from occurring in a subject that can be predisposed to the disease but has not yet been diagnosed as

having it; (ii) inhibiting the disease, i.e., arresting its development; or (iii) relieving the disease, i.e., causing regression of the disease.

[154] As used herein, the term “diagnosed” means having been subjected to a physical examination by a person of skill, for example, a physician, and found to have a condition that can be diagnosed or treated by the compounds, compositions, or methods disclosed herein.

[155] As used herein, the phrase “identified to be in need of treatment for a disorder,” or the like, refers to selection of a subject based upon need for treatment of the disorder. For example, a subject can be identified as having a need for treatment of a disorder (e.g., diabetes, or pre-diabetes, or a skin disease or disorder, or an inflammatory disease or disorder, or heart disease or disorder) based upon an earlier diagnosis by a person of skill and thereafter subjected to treatment for the disorder. It is contemplated that the identification can, in one aspect, be performed by a person different from the person making the diagnosis. It is also contemplated, in a further aspect, that the administration can be performed by one who performed the diagnosis.

[156] As used herein, the terms “administering” and “administration” refer to any method of providing a composition, complex, or a pharmaceutical preparation to a subject. Such methods are well known to those skilled in the art and include, but are not limited to: oral administration, transdermal administration, administration by inhalation, nasal administration, topical administration, intravaginal administration, ophthalmic administration, intraaural administration, intracerebral administration, rectal administration, sublingual administration, buccal administration, and parenteral administration, including injectable such as intravenous administration, intra-arterial administration, intramuscular administration, and subcutaneous administration. Administration can be continuous or intermittent. In various aspects, a preparation can be administered therapeutically; that is, administered to treat an existing disease or condition. In further various aspects, a preparation can be administered prophylactically; that is, administered for prevention of a disease or condition. In an aspect, the skilled person can determine an efficacious dose, an efficacious schedule, and an efficacious route of administration for a disclosed composition or a disclosed complex so as to treat a subject or inhibit or prevent an inflammatory reaction. In an aspect, the skilled person can also alter, change, or modify an aspect of an administering step so as to improve efficacy of a disclosed complex or disclosed composition.

[157] Throughout this application, various publications are referenced. The disclosures of these publications in their entireties are hereby incorporated by reference into this application in order to more fully describe the state of the art to which this pertains. The references disclosed are also individually and specifically incorporated by reference herein for the material contained in them that is discussed in the sentence in which the reference is relied upon.

[158] It is understood that the disclosed nucleic acids and proteins can be represented as a sequence consisting of the nucleotides of amino acids. There are a variety of ways to display these sequences, for example the nucleotide guanosine can be represented by G or g. Likewise the amino acid valine can be represented by Val or V. Those of skill in the art understand how to display and express any nucleic acid or protein sequence in any of the variety of ways that exist, each of which is considered herein disclosed. Specifically contemplated herein is the display of these sequences on computer readable mediums, such as, commercially available floppy disks, tapes, chips, hard drives, compact disks, and video disks, or other computer readable mediums. Also disclosed are the binary code representations of the disclosed sequences. Those of skill in the art understand what computer readable mediums. Thus, computer readable mediums on which the nucleic acids or protein sequences are recorded, stored, or saved.

[159] Disclosed are computer readable mediums comprising the sequences and information regarding the sequences set forth herein. Also disclosed are computer readable mediums comprising the sequences and information regarding the sequences set forth herein.

[160] It will be apparent to those skilled in the art that various modifications and variations can be made in the present disclosure without departing from the scope or spirit of the disclosure. Other embodiments of the disclosure will be apparent to those skilled in the art from consideration of the specification and practice of the disclosure disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope and spirit of the disclosure being indicated by the following claims.

[161] It is understood that the disclosed method and compositions are not limited to the particular methodology, protocols, and reagents described as these may vary. It is also to be understood that the terminology used herein is for the purpose of describing particular

embodiments only, and is not intended to limit the scope of the present disclosure which will be limited only by the appended claims.

[162] Throughout the description and claims of this specification, the word “comprise” and variations of the word, such as “comprising” and “comprises,” means “including but not limited to,” and is not intended to exclude, for example, other additives, components, integers or steps. In some instances, comprise may also include “consists of” or “consisting of”.

[163] In an exemplary aspect, the methods and systems can be implemented on a computer 5801 as illustrated in FIG. 58 and described below. Similarly, the methods and systems disclosed can utilize one or more computers to perform one or more functions in one or more locations. FIG. 58 is a block diagram illustrating an exemplary operating environment 5800 for performing the disclosed methods. This exemplary operating environment 5800 is only an example of an operating environment and is not intended to suggest any limitation as to the scope of use or functionality of operating environment architecture. Neither should the operating environment 5800 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment 5800.

[164] The present methods and systems can be operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that can be suitable for use with the systems and methods comprise, but are not limited to, personal computers, server computers, laptop devices, and multiprocessor systems. Additional examples comprise set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that comprise any of the above systems or devices, and the like.

[165] The processing of the disclosed methods and systems can be performed by software components. The disclosed systems and methods can be described in the general context of computer-executable instructions, such as program modules, being executed by one or more computers or other devices. Generally, program modules comprise computer code, routines, programs, objects, components, data structures, and/or the like that perform particular tasks or implement particular abstract data types. The disclosed methods can also be practiced in grid-based and distributed computing environments where tasks are performed by remote

processing devices that are linked through a communications network. In a distributed computing environment, program modules can be located in local and/or remote computer storage media including memory storage devices.

[166] Further, one skilled in the art will appreciate that the systems and methods disclosed herein can be implemented via a general-purpose computing device in the form of a computer 5801. The computer 5801 can comprise one or more components, such as one or more processors 5803, a system memory 5812, and a bus 5813 that couples various components of the computer 5801 including the one or more processors 5803 to the system memory 5812. In the case of multiple processors 5803, the system can utilize parallel computing.

[167] The bus 5813 can comprise one or more of several possible types of bus structures, such as a memory bus, memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. By way of example, such architectures can comprise an Industry Standard Architecture (ISA) bus, a Micro Channel Architecture (MCA) bus, an Enhanced ISA (EISA) bus, a Video Electronics Standards Association (VESA) local bus, an Accelerated Graphics Port (AGP) bus, and a Peripheral Component Interconnects (PCI), a PCI-Express bus, a Personal Computer Memory Card Industry Association (PCMCIA), Universal Serial Bus (USB) and the like. The bus 5813, and all buses specified in this description can also be implemented over a wired or wireless network connection and one or more of the components of the computer 5801, such as the one or more processors 5803, a mass storage device 5804, an operating system 5805, SVM software 5806, SVM-based data 5807, a network adapter 5808, system memory 5812, an Input/Output Interface 5810, a display adapter 5809, a display device 5811, and a human machine interface 5802, can be contained within one or more remote computing devices 5814a,b,c at physically separate locations, connected through buses of this form, in effect implementing a fully distributed system.

[168] The computer 5801 typically comprises a variety of computer readable media. Exemplary readable media can be any available media that is accessible by the computer 5801 and comprises, for example and not meant to be limiting, both volatile and non-volatile media, removable and non-removable media. The system memory 5812 can comprise computer readable media in the form of volatile memory, such as random access memory (RAM), and/or non-volatile memory, such as read only memory (ROM). The system

memory 5812 typically can comprise data such as SVM-based data 5807 and/or program modules such as operating system 5805 and SVM software 5806 that are accessible to and/or are operated on by the one or more processors 5803.

[169] In another aspect, the computer 5801 can also comprise other removable/non-removable, volatile/non-volatile computer storage media. The mass storage device 5804 can provide non-volatile storage of computer code, computer readable instructions, data structures, program modules, and other data for the computer 5801. For example, a mass storage device 5804 can be a hard disk, a removable magnetic disk, a removable optical disk, magnetic cassettes or other magnetic storage devices, flash memory cards, CD-ROM, digital versatile disks (DVD) or other optical storage, random access memories (RAM), read only memories (ROM), electrically erasable programmable read-only memory (EEPROM), and the like.

[170] Optionally, any number of program modules can be stored on the mass storage device 5804, including by way of example, an operating system 5805 and SVM software 5806. One or more of the operating system 5805 and SVM software 5806 (or some combination thereof) can comprise elements of the programming and the SVM software 5806. SVM-based data 5807 can also be stored on the mass storage device 5804. SVM-based data 5807 can be stored in any of one or more databases known in the art. Examples of such databases comprise, DB2®, Microsoft® Access, Microsoft® SQL Server, Oracle®, mySQL, PostgreSQL, and the like. The databases can be centralized or distributed across multiple locations within the network 5815.

[171] In another aspect, the user can enter commands and information into the computer 5801 via an input device (not shown). Examples of such input devices comprise, but are not limited to, a keyboard, pointing device (*e.g.*, a computer mouse, remote control), a microphone, a joystick, a scanner, tactile input devices such as gloves, and other body coverings, motion sensor, and the like. These and other input devices can be connected to the one or more processors 5803 via a human machine interface 5802 that is coupled to the bus 5813, but can be connected by other interface and bus structures, such as a parallel port, game port, an IEEE 1394 Port (also known as a Firewire port), a serial port, network adapter 5808, and/or a universal serial bus (USB).

[172] In yet another aspect, a display device 5811 can also be connected to the bus 5813 via an interface, such as a display adapter 5809. It is contemplated that the computer 5801 can have more than one display adapter 5809 and the computer 5801 can have more than one display device 5811. For example, a display device 5811 can be a monitor, an LCD (Liquid Crystal Display), light emitting diode (LED) display, television, smart lens, smart glass, and/or a projector. In addition to the display device 5811, other output peripheral devices can comprise components such as speakers (not shown) and a printer (not shown) which can be connected to the computer 5801 via Input/Output Interface 5810. Any step and/or result of the methods can be output in any form to an output device. Such output can be any form of visual representation, including, but not limited to, textual, graphical, animation, audio, tactile, and the like. The display 5811 and computer 5801 can be part of one device, or separate devices.

[173] The computer 5801 can operate in a networked environment using logical connections to one or more remote computing devices 5814a,b,c. By way of example, a remote computing device 5814a,b,c can be a personal computer, computing station (e.g., workstation), portable computer (e.g., laptop, mobile phone, tablet device), smart device (e.g., smartphone, smart watch, activity tracker, smart apparel, smart accessory), security and/or monitoring device, a server, a router, a network computer, a peer device, edge device or other common network node, and so on. Logical connections between the computer 5801 and a remote computing device 5814a,b,c can be made via a network 5815, such as a local area network (LAN) and/or a general wide area network (WAN). Such network connections can be through a network adapter 5808. A network adapter 5808 can be implemented in both wired and wireless environments. Such networking environments are conventional and commonplace in dwellings, offices, enterprise-wide computer networks, intranets, and the Internet.

[174] For purposes of illustration, application programs and other executable program components such as the operating system 5805 are illustrated herein as discrete blocks, although it is recognized that such programs and components can reside at various times in different storage components of the computing device 5801, and are executed by the one or more processors 5803 of the computer 5801. An implementation of SVM software 5806 can be stored on or transmitted across some form of computer readable media. Any of the disclosed methods can be performed by computer readable instructions embodied on

computer readable media. Computer readable media can be any available media that can be accessed by a computer. By way of example and not meant to be limiting, computer readable media can comprise “computer storage media” and “communications media.” “Computer storage media” can comprise volatile and non-volatile, removable and non-removable media implemented in any methods or technology for storage of information such as computer readable instructions, data structures, program modules, or other data. Exemplary computer storage media can comprise RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by a computer.

[175] The methods and systems can employ artificial intelligence (AI) techniques such as machine learning and iterative learning. Examples of such techniques include, but are not limited to, expert systems, case based reasoning, Bayesian networks, behavior based AI, neural networks, fuzzy systems, evolutionary computation (e.g. genetic algorithms), swarm intelligence (e.g. ant algorithms), and hybrid intelligent systems (e.g. Expert inference rules generated through a neural network or production rules from statistical learning).

[176] Those skilled in the art will recognize, or be able to ascertain using no more than routine experimentation, many equivalents to the specific embodiments of the method and compositions described herein. Such equivalents are intended to be encompassed by the following claims.

REFERENCES

1. Bailey T, Elkan C. 1994. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc Int Conf Intell Syst Mol Biol* 2: 28–36.
2. Banerji J. 1981. Expression of a α -globin gene is enhanced by remote SV40 DNA sequences. *Cell* 27: 299–308.
3. Beer MA, Tavazoie S. 2004. Predicting gene expression from sequence. *Cell* 117: 185–198.
4. Ben-Hur A, Ong CS, Sonnenburg S, Schölkopf B, Rätsch G. 2008. Support vector machines and kernels for computational biology. *PLoS Comput Biol* 4: e1000173. doi: 10.1371/journal.pcbi.1000173.
5. Berger MF, Badis G, Gehrke AR, Talukder S, Philippakis AA, Peña-Castillo L, Alleyne TM, Mnaimneh S, Botvinnik OB, Chan ET et al. 2008. Variation in homeodomain DNA binding revealed by high-resolution analysis of sequence preferences. *Cell* 133: 1266–1276.
6. Bertrand N, Castro DS, Guillemot F. 2002. Proneural genes and the specification of neural cell types. *Nat Rev Neurosci* 3: 517–530.
7. Blackwood EM, Kadonaga JT. 1998. Going the distance: A current view of enhancer action. *Science* 281: 60–63.
8. Boser BE, Guyon IM, Vapnik VN. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. Association for Computing Machinery (ACM), New York.
9. Bryne JC, Valen E, Tang ME, Marstrand T, Winther O, da Piedade I, Krogh A, Lenhard B, Sandelin A. 2008. JASPAR, the open access database of transcription factor-binding profiles: New content and tools in the 2008 update. *Nucleic Acids Res* 36: D102–D106.

10. Bulfone A, Puelles L, Porteus M, Frohman M, Martin G, Rubenstein J. 1993. Spatially restricted expression of *Dlx-1*, *Dlx-2* (*Tes-1*), *Gbx-2*, and *Wnt-3* in the embryonic day 12.5 mouse forebrain defines potential transverse and longitudinal segmental boundaries. *J Neurosci* 13: 3155–3172.
11. Carter D, Chakalova L, Osborne CS, Dai Y, Fraser P. 2002. Long-range chromatin regulatory interactions *in vivo*. *Nat Genet* 32: 623–626.
12. Chan HM, La Thangue NB. 2001. P300/CBP proteins: HATs for transcriptional bridges and scaffolds. *J Cell Sci* 114: 2363–2373.
13. Chen X, Xu H, Yuan P, Fang F, Huss M, Vega VB, Wong E, Orlov YL, Zhang W, Jiang J, et al. 2008. Integration of external signaling pathways with the core transcriptional network in embryonic stem cells. *Cell* 133: 1106–1117.
14. Demarque M, Spitzer NC. 2010. Activity-dependent expression of *Lmx1b* regulates specification of serotonergic neurons modulating swimming behavior. *Neuron* 67: 321–334.
15. Elnitski L, Hardison RC, Li J, Yang S, Kolbe D, Eswara P, O'Connor MJ, Schwartz S, Miller W, Chiaromonte F. 2003. Distinguishing regulatory DNA from neutral sites. *Genome Res* 13: 64–72.
16. ENCODE Project Consortium. 2007. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature* 447: 799–816.
17. Erives A, Levine M. 2004. Coordinate enhancers share common organizational features in the *Drosophila* genome. *Proc Natl Acad Sci* 101: 3851–3856.
18. Fisher S, Grice EA, Vinton RM, Bessling SL, McCallion AS. 2006. Conservation of RET regulatory function from human to zebrafish without sequence similarity. *Science* 312: 276–279.

19. Flavell SW, Greenberg ME. 2008. Signaling mechanisms linking neuronal activity to gene expression and plasticity of the nervous system. *Annu Rev Neurosci* 31: 563–590.
20. Frietze S, Lan X, Jin VX, Farnham PJ. 2010. Genomic targets of the KRAB and SCAN domain-containing zinc finger protein 263. *J Biol Chem* 285: 1393–1403.
21. Furey TS, Cristianini N, Duffy N, Bednarski DW, Schummer M, Haussler D. 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* 16: 906–914.
22. Ghanem N, Jarinova O, Amores A, Long Q, Hatch G, Park BK, Rubenstein JLR, Ekker M. 2003. Regulatory roles of conserved intergenic domains in vertebrate Dlx bigene clusters. *Genome Res* 13: 533–543.
23. Gotea V, Visel A, Westlund JM, Nobrega MA, Pennacchio LA, Ovcharenko I. 2010. Homotypic clusters of transcription factor binding sites are a key component of human promoters and enhancers. *Genome Res* 20: 565–577.
24. Gupta S, Stamatoyannopoulos JA, Bailey TL, Noble W. 2007. Quantifying similarity between motifs. *Genome Biol* 8: R24. doi: 10.1186/gb-2007-8-2-r24.
25. Hallikas O, Palin K, Sinjushina N, Rautiainen R, Partanen J, Ukkonen E, Taipale J. 2006. Genome-wide prediction of mammalian enhancers based on analysis of transcription- factor binding affinity. *Cell* 124: 47–59.
26. Heintzman ND, Stuart RK, Hon G, Fu Y, Ching CW, Hawkins RD, Barrera LO, Van Calcar S, Qu C, Ching KA, et al. 2007. Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome. *Nat Genet* 39: 311–318.
27. Heintzman ND, Hon GC, Hawkins RD, Kheradpour P, Stark A, Harp LF, Ye Z, Lee LK, Stuart RK, Ching CW, et al. 2009. Histone modifications at human enhancers reflect global cell-type-specific gene expression. *Nature* 459: 108–112.

28. Hughes JD, Estep PW, Tavazoie S, Church GM. 2000. Computational identification of Cis-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. *J Mol Biol* 296: 1205–1214.
29. Joachims T. 1999. Making large-scale support vector machine learning practical. In *Advances in kernel methods*, pp. 169–184. MIT Press, Cambridge, MA.
30. John S, Sabo PJ, Thurman RE, Sung M-H, Biddie SC, Johnson TA, Hager GL, Stamatoyannopoulos JA. 2011. Chromatin accessibility pre-determines glucocorticoid receptor binding patterns. *Nat Genet* 43: 264–268.
31. Johnson DS, Mortazavi A, Myers RM, Wold B. 2007. Genome-wide mapping of *in vivo* protein-DNA interactions. *Science* 316: 1497–1502.
32. Kadonaga JT. 2004. Regulation of RNA polymerase II transcription by sequence-specific DNA binding factors. *Cell* 116: 247–257.
33. Karchin R, Karplus K, Haussler D. 2002. Classifying G-protein coupled receptors with support vector machines. *Bioinformatics* 18: 147–159.
34. Karolchik D, Kuhn RM, Baertsch R, Barber GP, Clawson H, Diekhans M, Giardine B, Harte RA, Hinrichs AS, Hsu F, et al. 2008. The UCSC Genome Browser Database: 2008 update. *Nucleic Acids Res* 36: D773–D779.
35. Keilwagen J, Grau J, Paponov IA, Posch S, Strickert M, Grosse I. 2011. De-novo discovery of differentially abundant transcription factor binding sites including their positional preference. *PLoS Comput Biol* 7: e1001070. doi: 10.1371/journal.pcbi.1001070.
36. Kim T, Hemberg M, Gray JM, Costa AM, Bear DM, Wu J, Harmin DA, Laptewicz M, Barbara-Haley K, Kuersten S, et al. 2010. Widespread transcription at neuronal activity-regulated enhancers. *Nature* 465: 182–187.
37. King DC, Taylor J, Elnitski L, Chiaromonte F, Miller W, Hardison RC. 2005. Evaluation of regulatory potential and conservation scores for detecting cis-regulatory modules in aligned mammalian genome sequences. *Genome Res* 15: 1051–1060.

38. Koh K, Kim S-J, Boyd S. 2007. An interior-point method for large-scale l1-regularized logistic regression. *J Mach Learn Res* 8: 1519–1555.
39. Kurokawa D, Kiyonari H, Nakayama R, Kimura-Yoshida C, Matsuo I, Aizawa S. 2004. Regulation of Otx2 expression and its functions in mouse forebrain and midbrain. *Development* 131: 3319–3331.
40. Lee, et al., *Genome Res.* 2011. 21: 2167-2180, and supplemental material are at <http://www.genome.org/cgi/doi/10.1101/gr.121905.111>, each of which is herein incorporated in its entirety.
41. Lee JE. 1997. Basic helix-loop-helix genes in neural development. *Curr Opin Neurobiol* 7: 13–20.
42. Leslie C, Eskin E, Noble WS. 2002. The spectrum kernel: A string kernel for SVM protein classification. *Pac Symp Biocomput* 7: 564–575.
43. Leslie C, Eskin E, Cohen A, Weston J, Noble WS. 2004. Mismatch string kernels for discriminative protein classification. *Bioinformatics* 20: 467–476.
44. Leung G, Eisen MB. 2009. Identifying cis-regulatory sequences by word profile similarity. *PLoS ONE* 4: e6901. doi: 10.1371/journal.pone.0006901.
45. Lin, H.T., Lin, C.J. and Weng, R.C. 2003. A note on Platt's probabilistic outputs for support vector machines machine learning. *Mach. Learn.* 68: 267-276.
46. Maniatis T, Falvo JV, Kim TH, Kim TK, Lin CH, Parekh BS, Watheliet MG. 1998. Structure and function of the interferon- γ enhanceosome. *Cold Spring Harb Symp Quant Biol* 63: 609–620.
47. Mason S, Piper M, Gronostajski RM, Richards LJ. 2008. Nuclear factor one transcription factors in CNS development. *Mol Neurobiol* 39: 10–23.
48. Matsuo I, Kuratani S, Kimura C, Takeda N, Aizawa S. 1995. Mouse Otx2 functions in the formation and patterning of rostral head. *Genes Dev* 9: 2646–2658.

49. Matys V, Fricke E, Geffers R, Gossling E, Haubrock M, Hehl R, Hornischer K, Karas D, Kel AE, Kel-Margoulis OV, et al. 2003. TRANSFAC(R): Transcriptional regulation, from patterns to profiles. *Nucleic Acids Res* 31: 374–378.
50. McGaughey DM, Vinton RM, Huynh J, Al-Saif A, Beer MA, McCallion AS. 2008. Metrics of sequence constraint overlook regulatory sequences in an exhaustive analysis at phox2b. *Genome Res* 18: 252–260.
51. Megraw M, Pereira F, Jensen ST, Ohler U, Hatzigeorgiou AG. 2009. A transcription factor affinity-based code for mammalian transcription initiation. *Genome Res* 19: 644–656.
52. Meinicke P, Tech M, Morgenstern B, Merkl R. 2004. Oligo kernels for datamining on biological sequences: A case study on prokaryotic translation initiation sites. *BMC Bioinformatics* 5: 169. doi: 10.1186/1471-2105-5-169.
53. Narlikar L, Sakabe NJ, Blanski AA, Arimura FE, Westlund JM, Nobrega MA, Ovcharenko I. 2010. Genome-wide discovery of human heart enhancers. *Genome Res* 20: 381–392.
54. Newburger DE, Bulyk ML. 2009. UniPROBE: An online database of protein binding microarray data on protein-DNA interactions. *Nucleic Acids Res* 37: D77–D82.
54. Noonan JP, McCallion AS. 2010. Genomics of long-range regulatory elements. *Annu Rev Genomics Hum Genet* 11: 1–23.
55. Patel, M., Simon, J., Iglesia, M., Wu, S.B., McFadden, A., Lieb, J.D. and Davis, I.J. 2012. Tumor-specific retargeting of an oncogenic transcription factor chimera results in dysregulation of chromatic and transcription. *Genome Res* 22: 259-270.
56. Pavese G, Mauri G, Pesole G. 2001. An algorithm for finding signals of unknown length in DNA sequences. *Bioinformatics* 17 (Suppl 1): S207–S214.
57. Peckham HE, Thurman RE, Fu Y, Stamatoyannopoulos JA, Noble WS, Struhl K, Weng Z. 2007. Nucleosome positioning signals in genomic DNA. *Genome Res* 17: 1170–1177.

58. Pennacchio LA, Ahituv N, Moses AM, Prabhakar S, Nobrega MA, Shoukry M, Minovitsky S, Dubchak I, Holt A, Lewis KD, et al. 2006. *In vivo* enhancer analysis of human conserved noncoding sequences. *Nature* 444: 499–502.
59. Platt, J.C. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Smola, A., Bartlett, P., Scholkopf, B. and Schuurmans, D. (eds). *Advances in Large Margin Classifiers*. MIT Press, Cambridge, MA: 67-74.
60. Räscher G, Sonnenburg S, Schölkopf B. 2005. RASE: Recognition of alternatively spliced exons in *C. elegans*. *Bioinformatics* 21 (Suppl 1): i369–i377.
61. Robertson G, Hirst M, Bainbridge M, Bilenky M, Zhao Y, Zeng T, Euskirchen G, Bernier B, Varhol R, Delaney A, et al. 2007. Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing. *Nat Methods* 4: 651–657.
62. Ross SE, Greenberg ME, Stiles CD. 2003. Basic helix-loop-helix factors in cortical development. *Neuron* 39: 13–25.
63. Schölkopf B, Tsuda K, Vert JP. 2004. *Kernel methods in computational biology*. MIT Press, Cambridge, MA.
64. Schultheiss SJ. 2010. Kernel-based identification of regulatory modules. In *Computational biology of transcription factor binding* (ed. Ladunga I.), Vol. 674, pp. 213–223. Humana Press, Totowa, NJ.
65. Schultheiss SJ, Busch W, Lohmann JU, Kohlbacher O, Räscher G. 2009. KIRMES: Kernel-based identification of regulatory modules in euchromatic sequences. *Bioinformatics* 25: 2126–2133.
66. Siepel A, Bejerano G, Pedersen JS, Hinrichs AS, Hou M, Rosenbloom K, Clawson H, Spieth J, Hillier LW, Richards S, et al. 2005. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome Res* 15: 1034–1050.

67. Sonnenburg S, Rätsch G, Schäfer C, Schölkopf B. 2006a. Large scale multiple kernel learning. *J Mach Learn Res* 7: 1531–1565.
68. Sonnenburg S, Zien A, Ratsch G. 2006b. ARTS: Accurate recognition of transcription starts in human. *Bioinformatics* 22: e472–e480.
69. Sonnenburg S, Schweikert G, Philips P, Behr J, Ratsch G. 2007. Accurate splice site prediction using support vector machines. *BMC Bioinformatics* 8: S7. doi: 10.1186/1471-2105-8-S10-S7.
68. Sonnenburg S, Zien A, Philips P, Ratsch G. 2008. POIMs: positional oligomer importance matrices—understanding support vector machine-based signal detectors. *Bioinformatics* 24: i6–i14.
69. Storey JD, Tibshirani R. 2003. Statistical significance for genomewide studies. *Proc Natl Acad Sci* 100: 9440–9445.
70. Su J, Teichmann SA, Down TA. 2010. Assessing computational methods of cis-regulatory module prediction. *PLoS Comput Biol* 6: e1001020. doi: 10.1371/journal.pcbi.1001020.
71. Thanos D, Maniatis T. 1995. Virus induction of human IFN γ gene expression requires the assembly of an enhanceosome. *Cell* 83: 1091–1100.
72. Vandewalle C, Roy F, Berx G. 2008. The role of the ZEB family of transcription factors in development and disease. *Cell Mol Life Sci* 66: 773–787.
73. Vapnik VN. 1995. *The nature of statistical learning theory*. Springer, New York.
74. Visel A, Prabhakar S, Akiyama JA, Shoukry M, Lewis KD, Holt A, Plajzer-Frick I, Afzal V, Rubin EM, Pennacchio LA. 2008. Ultraconservation identifies a small subset of extremely constrained developmental enhancers. *Nat Genet* 40: 158–160.
75. Visel A, Blow MJ, Zhang T, Akiyama JA, Holt A, Plajzer-Frick I, Shoukry M, Wright C, Chen F, Afzal V, et al. 2009. ChIP-seq accurately predicts tissue-specific activity of enhancers. *Nature* 457: 854–858.

76. Wigle JT, Eisenstat DD. 2008. Homeobox genes in vertebrate forebrain development and disease. *Clin Genet* 73: 212–226.
77. Wilson M, Koopman P. 2002. Matching SOX: Partner proteins and cofactors of the SOX family of transcriptional regulators. *Curr Opin Genet Dev* 12: 441–446.
78. Wilson MD, Barbosa-Morais NL, Schmidt D, Conboy CM, Vanes L, Tybulewicz VLJ, Fisher EMC, Tavaré S, Odom DT. 2008. Species-specific transcription in mice carrying human chromosome 21. *Science* 322: 434–438.
79. Woolfe A, Goodson M, Goode DK, Snell P, McEwen GK, Vavouri T, Smith SF, North P, Callaway H, Kelly K, et al. 2004. Highly conserved noncoding sequences are associated with vertebrate development. *PLoS Biol* 3: e7. doi: 10.1371/journal.pbio.0030007.
80. Zerucha T, Stühmer T, Hatch G, Park BK, Long Q, Yu G, Gambarotta A, Schultz JR, Rubenstein JLR, Ekker M. 2000. A highly conserved enhancer in the *Dlx5/Dlx6* intergenic region is the site of cross-regulatory interactions between *Dlx* genes in the embryonic forebrain. *J Neurosci* 20: 709–721.
81. Fletz-Brant, Christopher, et al., kmer-SVM: a web server for identifying predictive regulatory sequence features in genomic data sets, *Nucleic Acids Res.* 2013, Vol. 41, doi:10.1093/nar/gkt519.
82. Gorkin, et al., *Genome Res.* 2012, 22:2290-2301, Integration of ChIP-eq and machine learning reveals enhancers and a predictive regulatory sequence vocabulary in melanocytes.
83. Ghandi, et al., Robust k-mer frequency estimation using gapped k-mers, *J. Math.Biol.* DOI 10.1007/s00285-013-0705-3.

EXAMPLES

EXAMPLE 1*Methods Data Sets*

[177] As positive data sets, initially the genome-wide in vivo EP300 binding sites identified by ChIP-seq (Visel et al. 2009) were used, composed of three different sets of tissue-specific enhancers (forebrain, midbrain, and limb) of embryonic day 11.5 mouse embryos. There were 2453, 561, and 2105 sites reported, respectively, and the entire sequences were directly used without modification. Two other data sets were analyzed (Chen et al. 2008; Kim et al. 2010). Chen et al. reported 524 EP300 binding sites in mouse embryonic stem cells, and Kim et al. reported ~12,000 neural activity-dependent CREBBP binding sites in stimulated cultured mouse cortical neurons. Since both CREBBP data sets report only peaks of the ChIP-seq signals, to obtain sequences for further analysis, an extension of 100 bp (Fig. 3G) or 400 bp (Fig. 3H) in both directions from these peaks was made.

[178] Negative sequence sets were generated to match the distribution of sequence length and repeat element fraction of the corresponding positive sets (Fig. 11). Repeat fractions were calculated using the repeat masked sequence data from the UCSC genome browser (Karolchik et al. 2008). random genomic sequences were selected from the mouse genome according to the following rejection sampling algorithm:

1. Sample a length l from the enhancer length distribution.
2. Sample a sequence of the length l , randomly from the genome.
3. Let x be the repeat fraction of the sampled sequence. Sample $Y \sim \text{Bernoulli}(\alpha(x)lq(x))$, where $p(x)$ is the probability that x occurs in the enhancers, $q(x)$ is the probability that x occurs in the genomic sequence, α is the constant so that the maximum of $p(x)lq(x)$ equals 1.
4. Accept the sequence if $Y = 1$; reject otherwise.
5. Repeat 1–4 until the desired number of sequences are sampled.

[179] All positive and negative sequence data sets used for this analysis are available at <http://www.beerlab.org/p300enhancer>. The following negative set sizes were used—EP300 f

b: n = 4000, 2453 (1×), 122,650 (50×), 245,300 (100×); EP300 mb: n = 4000, 561 (1×), 28,050 (50×), 56,100 (100×); EP300 lb: n = 4000, 2105 (1×), 105,250 (50×), 210,500 (100×); EP300 f b human: n = 2192 (1×); EP300 ES: n = 524 (1×), 5240 (10×), 26,200 (50×), 52,400 (100×); CREBBP neuron: n = 11,847 (1×), 592,350 (50×), 1,184,700 (100×); ZNF263: n = 1418 (1×), 70,900 (50×), 141,800 (100×).

Support vector machine

[180] An SVM (Boser et al. 1992; Vapnik 1995) finds a decision boundary that separates the positive and negative training data. This decision boundary is a hyperplane which maximizes the margin between the two sets in the feature vector space. Used were N labeled vectors (x_i, γ_i) , $i = 1, \dots, N$, where $x_i \in \mathbb{R}^n$ and $\gamma_i \in \{+1, -1\}$ is the class label. For the linear case, the decision boundary is found by minimizing $\|w\|^2$ such that $\gamma_i(x_i \cdot w + b) \geq 1$, $i = 1, \dots, N$. In practice, the optimal solution was found by maximizing the dual form: $\sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \gamma_i \alpha_i \gamma_j \alpha_j (x_i \cdot x_j)$ over α_j with the constraints, $\alpha_i \geq 0$, and $\sum_{i=1}^N \alpha_i \gamma_i = 0$ (Joachims 1999; Sonnenburg et al. 2006a). The SVM weight vector w was constructed from the α_i , using $w = \sum_{i=1}^N \gamma_i \alpha_i x_i$. The SVM discriminant function, or “SVM score,” $f_{svm}(x) = w \cdot x + b = \sum_{i=1}^N \gamma_i \alpha_i (x_i \cdot x) + b$, represented the distance of any vector x from the decision boundary, and determines the predicted label of the vector x .

[181] The inner product $(x_i \cdot x_j)$ was a measure of the similarity of any two data points i and j in the feature space. The generality of the SVM arose from the fact that this term may be replaced by a more general measure of similarity, a kernel function $K(x_i, x_j)$. Different kernels refer to different methods of measuring similarity. A general measure of sequence similarity is the k -spectrum kernel (Leslie et al. 2002), which describes the similarity of k -mer frequencies of two sequences. This kernel produced the best results in the present method, was easy to interpret, and can easily represent a combination of TF binding sites. To implement the k -spectrum kernel, a k -mer count vector was generated for the full set of distinct k -mers for each sequence. Then the count vector was normalized so that $\|x\| = 1$ to reduce the effect of the variable length of different enhancers. This normed vector is referred to as the “ k -mer frequency vector.” The kernel function was then the inner product between two normalized frequency vectors. To reflect the fact that TFs bind double stranded DNA, the spectrum kernel function was slightly modified to account for both orientations. Instead of counting only an exact k -mer, its reverse complement was also counted, and then

redundant k -mers were removed. For example, only one of AATGCT and AGCATT appears on the list of distinct k -mers. For 6-mers, there are 2080 distinct features after removing reverse complements; for 7-mers, there are 8192. This modification was applied to all kernel functions. The only difference between the k -spectrum kernel and the (k,m) -mismatch kernel is that the mismatch kernel allowed m mismatches when counting k -mers (Leslie et al. 2004), reflecting the fact that some TFs bind degenerate sites. The Gaussian kernel used the same feature vectors as the k -spectrum kernel but used a nonlinear similarity measure via the kernel function $K(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / 2\sigma^2)$. The method utilizes the Shogun machine learning toolbox (Sonnenburg et al. 2006a) and SVM light (Joachims 1999).

EXAMPLE 2

Details of auxiliary modules

Score sequences of interest

[182] Once the SVM is trained, in addition to classifying the CV test sets, a trained SVM can be used to score any sequence of interest. Although the rank of the SVM scores is significant, the scale of the SVM scores is generally not. Therefore, this SVM score may be converted into a probability that the element is positive, by reporting the posterior probability that each sequence is in the positive class, using the algorithm described in (45,59). For example, input may be a set of sequences in FASTA format and the outputs were the SVM score and posterior probability. Parameters to produce this posterior probability may be included in the weight table output of the trained SVM. Genome-wide predictions may also be made using the SVM methods disclosed herein by splitting a genome into chunks of a length c bp that overlap each other by v bp. The results may then be used as input for determining sequences of interest.

Sequence profiles

[183] As discussed earlier in the text, the sequence profiles, or distributions of length, GC content and repeat fraction content in the positive and negative sequences were matched. It may be useful to compare the sequence profiles of other sets of genomic intervals by calculating and reporting the sequence profile of the regions specified by these coordinates.

Kmer to MEME

[184] This step takes the output file of weights created by training a kmer-SVM and generates PWMs for kmers with the largest and smallest (most positive and most negative) weights. The user specifies how many kmers to be returned, with a maximum of 50. The output of this program is a MEME-formatted list of PWMs.

Tomtom

[185] To enable a user to visualize the kmers identified as predictive by kmer-SVM, a local instance of the Tomtom (15) program was implemented. Briefly, Tomtom searches databases of TF motifs for matches with input motifs by using column-wise similarity measures between PWMs. Users can create PWMs by converting Kmer output to MEME and using this as input for Tomtom. For measures of similarity, the Euclidean distance may be used, which can be thought of as the length of the straight line between two PWMs, the Pearson correlation coefficient, which measures the similarity between two PWMs, and the Sandelin–Wasserman function, which sums the column-wise differences between PWMs. Also the choice of E-value or q-value as scoring criteria may be used. The E-value controls the expected number of false positives and can be any number, whereas the q-value controls the false discovery rate and is a number between 0 and 1. Running Tomtom in the default configuration of the Pearson correlation coefficient as distance metric and the q-value as criteria is an optional step of disclosed methods.

EXAMPLE 3

[186] Regulatory control of gene expression in epidermal melanocytes, the pigment-producing cells that generate skin and hair color, was investigated. These cells also play a central role in several pathological phenotypes, including melanoma, albinism, and vitiligo (for review, see Lin and Fisher 2007). These qualities, along with extensive knowledge about the key TFs and developmental origins of melanocytes (Silver et al. 2006; Hou and Pavan 2008; Thomas and Erickson 2008), make this lineage an attractive model system for the study of enhancers. ChIP-seq for EP300 and H3K4me1 were employed to identify melanocyte enhancers genome-wide. A novel set of criteria was used that takes into account both EP300 and H3K4me1 to define a single set of putative enhancers, and validate these enhancers through a series of in silico, in vitro, and in vivo analyses. Having validated the identified enhancers, they were used as a training set for a machine learning algorithm, developing a

comprehensive vocabulary of 6-mers predictive of melanocyte enhancer function with power to predict additional melanocyte enhancers in the mouse and human genomes. Our data established an extensive body of knowledge about regulatory control in melanocytes, which is relevant to phenotypic variation and disease. Moreover, a comprehensive approach was demonstrated that integrates ChIP-seq and machine learning to discover lineage-dependent enhancers and reveal the sequence vocabulary underlying their function.

RESULTS

Previously characterized melanocyte enhancers are bound by EP300 and flanked by H3K4me1

[187] Sought to be identified was a large set of putative melanocyte enhancers from which a predictive sequence vocabulary could be derived. The enhancer identification was initiated by performing ChIP-seq for both EP300 and H3K4me1 in a line of immortalized melanocytes (melan-a) derived from Ink4a-Arf-null mice on a C57BL/6J background (Bennett et al. 1987; Sviderskaya et al. 2002). 3622 and 59,965 ChIP-seq peaks for EP300 and H3K4me1 were identified, respectively. Expected was a priori that both EP300 and H3K4me1 would be enriched at melanocyte enhancers loci, based on similar findings in other cell types (Barski et al. 2007; Heintzman et al. 2009; Visel et al. 2009a; Wang et al. 2009). Consistent with these observations, the presence of enrichment for these factors at previously characterized melanocyte enhancers was confirmed (Fig. 38A,B; Table 4). More specifically, it was observed that a central EP300 peak overlaps these enhancers and that this peak is flanked on both sides by strong H3K4me1 enrichment. To further assess the relationship between EP300 and H3K4me1 in melanocytes, the distribution of H3K4me1 ChIP-seq reads relative EP300 peaks genome-wide were examined. It was found that H3K4me1 enrichment flanking EP300 peaks is a striking genome-wide trend (Fig. 38C, D), similar to observations made in other cell types (Heintzman et al. 2007, 2009; Ghisletti et al. 2010).

A specific EP300/H3K4me1 ChIP-seq signature identifies melanocyte enhancers genome-wide

[188] To identify a finite set of putative enhancers, a genome-wide search was performed for loci bearing the signature observed at previously characterized melanocyte enhancers, i.e., at which an EP300 peak is flanked by H3K4me1 enrichment. First, a set of H3K4me1-

flanked regions at which adjacent H3K4me1 peaks are separated by between 100 and 1500 bp ($n = 21,189$) were identified. This distance of 100–1500 bp was chosen based on the range of intervals between adjacent H3K4me1 peaks at known melanocyte enhancers (Fig. 10). Next, all EP300 peaks that overlap an H3K4me1-flanked region were identified. This approach, represented schematically in Figure 39A, yields 2489 loci at which an EP300 peak falls in a region flanked by H3K4me1 peaks. Hereafter, these 2489 loci were referred to as “putative melanocyte enhancers” (Tables 7-10). These putative melanocyte enhancers included previously reported enhancers at *Tyr* and *Sox10* (Murisier et al. 2007; Antonellis et al. 2008), as well as novel enhancers at a number of other genes central to melanocyte biology, including *Mitf*, *Tyrp1*, *Kit*, and *Mclr* (Fig. 11). For downstream analysis, the summit of the EP300 peak were used as a surrogate for the center of a given enhancer, and where necessary, the boundaries of the EP300 peak were used as surrogates for the enhancer’s boundaries.

[189] Several additional lines of evidence supported the imputed function of these 2489 putative melanocyte enhancers. First, the putative melanocyte enhancers showed evolutionary sequence constraint (Fig. 39B), providing independent evidence of their functional significance. Second, these putative melanocyte enhancers were enriched for sequence motifs predicted to bind key melanocyte TFs, including SOX10 and MITF, as detected by DREME (Fig. 39C; Bailey 2011). Mutations in SOX10 and MITF in humans cause Waardenburg syndrome (WS), a pleiotropic neural crest disorder with characteristic pigmentary defects (SOX10 mutations cause WS type 2E OMIM:611584 and 4C OMIM:613266; MITF mutations cause WS type 2A OMIM:193510) (McKusick 1998; [http:// omim.org/](http://omim.org/)), and both TFs are involved in the pathogenesis of melanoma (Cronin et al. 2009; Harris et al. 2010). Third, analysis with GREAT (McLean et al. 2010) reveals that genes proximal to the putative melanocyte enhancers (within 50 kb; see GREAT methods) are significantly associated with Gene Ontology (GO) terms relevant to melanocyte biology, including melanoma, melanosome, pigmentation, and melanocyte differentiation (Table 1). Furthermore, using previously reported gene expression data for the melan-a line (Buac et al. 2009), it was found that putative melanocyte enhancers are enriched near the most highly expressed genes and depleted near genes that are not expressed at appreciable levels (39. 2D), reflecting the expected distribution of active melanocyte enhancers.

[190] Although the 2489 putative melanocyte enhancers were enriched within 100 kb of highly expressed genes, they were not enriched in a 1-kb window immediately adjacent to the transcription start site (TSS) of these genes (39, 2E). This suggested that the enhancers identified were truly distal-acting and included very few, if any, proximal promoter elements. In contrast, EP300 peaks that were not flanked by H3K4me1 are far more likely to overlap annotated TSSs (Fig. 41A). This trend was also true for additional cell types in which data are available from the ENCODE and modENCODE Project consortia (The ENCODE Project Consortium 2007; The modENCODE Project Consortium 2009). Furthermore, in these cell types the non-H3K4me1-flanked EP300 peaks show markedly higher levels of ChIP-seq enrichment for RNA polymerase II and the promoter-associated modification H3K4me3 (Fig. 41B). Consistent with these observations, several melan-a EP300 peaks were noted at the promoters of melanocyte-related genes that have H3K4me1 enrichment on one side (upstream of the TSS) but are not flanked (Fig. 12). Somewhat surprisingly, EP300 peaks that were not flanked by H3K4me1 also showed higher levels of binding for CTCF in the ENCODE cell types examined (Fig. 41B). CTCF plays a central role in the function of insulator elements (Bell et al. 1999) and in physical organization of chromatin (Phillips and Corces 2009). In further comparing H3K4me1- flanked and non-H3K4me1-flanked EP300 peaks, it was found that H3K4me1-flanked peaks have higher levels of EP300 enrichment than non-H3K4me1-flanked peaks ($P = 2.5 \times 10^{-5}$) (Fig. 41C).

[191] Collectively, these data showed that the set of 2489 candidate loci was highly enriched for bona fide melanocyte enhancers. By selecting only those EP300 peaks that overlap H3K4me1-flanked regions, a set of putative melanocyte enhancers with stronger EP300 binding that includes fewer regions containing sequence features of nonenhancer regulatory elements such as promoters and insulators were obtained. Importantly, these characteristics of this approach are particularly well suited to the creation of a training set from which key sequence features of enhancers can be extracted. Furthermore, these results add to a growing body of evidence linking EP300 and H3K4me1 to enhancer function and suggest the existence of functionally distinct subsets of EP300 peaks that can be distinguished to some extent by proximal histone modifications.

Identified melanocyte enhancers direct reporter expression in melanocytes *in vitro* and *in vivo*

[192] Given the evidence already supporting the role of the identified putative melanocyte enhancers in melanocyte regulatory control, next it was sought to validate their biological

activity in reporter assays. To this end, 50 putative enhancers were first selected at random from the full set of 2489 and each was analyzed its ability to direct expression of a luciferase reporter gene in the melan-a line. It was found that 86% (43/50) of enhancers tested increased reporter expression greater than threefold relative to the minimal promoter alone (Fig. 42A; Table 5). Moreover, 72% (36/50) of enhancers tested increased reporter expression more than fivefold, and 48% (24/50) increased expression more than 10-fold relative to the minimal promoter alone. As there was considerable variation in the activity of melanocyte enhancers in this assay, an additional 10 regions were tested as negative controls. These regions were matched to the putative enhancers in average size and GC content but did not have significant EP300 or H3K4me1 ChIP-seq enrichment. None of these negative control regions increased reporter expression more than threefold relative to promoter alone (Fig. 13A). As expected, the difference in reporter expression between putative enhancers and negative control regions was highly significant ($P = 9.6 \times 10^{-7}$ by two-tailed t-test) (Fig. 42B).

[193] Three previously characterized melanocyte enhancers were also assayed for reference, which directed expression at levels 11-fold, 42-fold, and 51-fold higher than the minimal promoter alone, respectively (Fig. 13B). However, it should be noted that these three enhancers are not directly comparable to these test sequences because the critical regions of these enhancers have been refined in previous studies. In this assay, a given enhancer will show highest activity when the amplified region contains the motifs critical for enhancer function with as little additional sequence as possible.

[194] To further validate the biological activity of the putative enhancers, the ability of a subset ($n = 10$) to appropriately direct melanocyte expression of a GFP reporter in vivo in transgenic zebrafish was tested. An established pipeline was used for analyzing putative enhancers in zebrafish (Fisher et al. 2006a,b; McGaughey et al. 2008; Prasad et al. 2011), which has previously been used to analyze melanocyte regulatory elements at Sox10 (Antonellis et al. 2008) and GPNMB (Loftus et al. 2009). The 10 putative enhancers tested were chosen at random from the 50 analyzed in vitro as described above. It was found that 70% (7/10) of enhancers tested directed GFP expression in the melanocytes of mosaic transgenic zebrafish (Table 6). The observed reporter expression was consistent with what has been seen previously when assaying melanocyte enhancers (Loftus et al. 2009) and was highly specific to melanocytes (Fig.14). Consistent expression was not observed in other tissues with any of the seven positive constructs, with two exceptions that result from

inherent artifacts of the assay: (1) Background GFP expression in the yolk (into which the construct is injected at day 0) is always observed; and (2) expression in skeletal muscle is often observed, which is likely caused by a cryptic regulatory sequence in the backbone of the reporter construct that was not located. One melanocyte-negative construct (putative enhancer 25) did drive consistent expression in ganglia of the peripheral nervous system (PNS). Interestingly, the PNS and melanocytes both arise from the neural crest during embryonic development.

[195] The results of these functional assays demonstrate that the majority of putative melanocyte enhancers can direct gene expression in melanocytes both in vitro and in vivo, providing strong additional evidence that the identified loci function as melanocyte enhancers.

Machine learning reveals sequence features that underlie melanocyte enhancer function

[196] To more thoroughly investigate the sequence composition of melanocyte enhancers, the putative enhancers identified by ChIP-seq were used as a training set for a supervised machine learning algorithm based on the statistical framework of a SVM (Lee et al. 2011). This approach as applied to embryonic mouse enhancers from other tissues is presented in detail by Lee et al. (2011). Briefly, the SVM finds an optimal decision boundary to distinguish the set of enhancers from random genomic regions using sequences of length k (k -mers) as features. Here, the putative melanocyte enhancers were used as positive sequences, a 50×larger set of random genomic regions as negative sequences, and the full set of 2080 distinct 6-mers as features. It was previously found that 6-mers and 7-mers are more informative in these analyses than are k -mers of other lengths, and 6-mers are preferred for robustness and ease of interpretation (Lee et al. 2011). SVM training assigned a weight, w , to each feature (6-mer), which determined its relative contribution to the decision boundary. The SVM discriminatory function, $f_{SVM}(x) = wx + b$, represented the distance of a sequence x from the decision boundary and determined the predicted class, enhancer or nonenhancer, of the sequence x . This approach, which is called the kmer-SVM classifier, has three major advantages: (1) It identifies the specific sequences recognized by TFs active in melanocytes and provides independent support for these putative melanocyte enhancers based on previously known biology; (2) it allows the identification of additional melanocyte enhancers outside the original set of 2489 putative enhancers; and (3) it allows an indirect assessment of the quality of these putative enhancer set based on its sequence properties.

[197] After training, the kmer-SVM classifier was assessed by its ability to accurately predict the class of reserved test sets via five- fold cross validation, as shown by the area under (au) the receiver operating characteristic curve (ROC) and precision-recall curves (PRCs). The kmer-SVM trained on putative melanocyte enhancers achieved auROC of 0.912 and auPRC of 0.297, providing independent verification of the quality of the experimental enhancer identification.

[198] A feature of the kmer-SVM was that it produced a list of features, in this case all unique 6-mers ($n = 2080$) and the corresponding weight assigned to each feature by the SVM. The SVM weight represents the relative contribution of a given 6-mer to the overall predictive power of the classifier. Collectively, the list of weighted 6-mers provides a sequence vocabulary that is useful in interpreting the primary sequence of melanocyte enhancers. Importantly, the most predictive 6-mers (i.e., those assigned the largest SVM weights) correspond to binding sites for TFs known to be directly involved in melanocyte biology, including MITF, SOX10, and FOS/JUN (Fig. 15). These 6-mers, and the 6-mer predicted to bind TEAD1, are in agreement with motifs found by DREME to be enriched in the training set (see Fig. 39C). It is also notable that one of the top 6-mers (ranked fourth) is predicted to bind PAX3, a key regulator of melanocyte differentiation (Lang et al. 2005) which can cause Waardenburg syndrome type 1 and type 3 when mutated (OMIM:193500 and OMIM:148820, respectively). In addition, CREB1, SOX5, and RUNX-family TFs (predicted to bind 6-mers ranked fifth, eighth, and ninth, respectively) have been shown to play roles in regulating gene expression in melanocytes (Tada et al. 2002; Raveh et al. 2005; Saha et al. 2006; Kingo et al. 2008; Stolt et al. 2008; Kanaykina et al. 2010; Mizutani et al. 2010).

Sequenced-based predictions identify additional enhancers in the mouse and human genomes

[199] Having trained the kmer-SVM classifier, it was next sought to determine whether it could be used to predict additional melanocyte enhancers genome-wide from primary sequence alone. Though these computational predictions are not likely to be as accurate as ChIP-seq, demonstrating that the kmer-SVM can predict bona fide enhancers is a powerful validation of the sequence vocabulary of weighted 6-mers on which the predictions are based. Furthermore, the ability to make enhancer predictions from sequence is particularly useful in genomes for which ChIP-seq data are not readily available. To make enhancer predictions

genome-wide, first the mouse genome was segmented into 400-bp regions with 300 bp overlap and scored all regions with the kmer-SVM. The top 10,000 regions were chosen for further analysis, corresponding to an SVM cut-off score of 1.0 and yielding a precision of 0.74 and recall of .05 estimated from the PR curve. Any predicted regions overlapping the original training set were then eliminated (508 regions overlapping 348 enhancers from the original training set) and any overlapping regions were merged. None of the six previously characterized melanocyte enhancers in Table 4 overlap a kmer-SVM prediction, though it should be noted that four are included in the training set as they were bound by EP300 and flanked by H3K4me1 (Tyr DRE-15kb, Sox10 MCS4, Sox10 MCS5, Sox10 MCS9).

[200] Ultimately, a set of 7361 predicted melanocyte enhancers (Table11) were obtained. These predicted enhancers showed strong sequence constraint (Fig. 7A), albeit to a lesser extent than the original set of putative enhancers. In addition, the predicted enhancers also showed an EP300 and H3K4me1 ChIP-seq signature reminiscent of the original enhancer set (Fig. 7B). This suggests that the kmer-SVM predictions shared underlying biology with the original set of 2489 putative enhancers, though the ChIP- seq signal at these loci was much lower than at regions detected by peak calling (Fig.16). Further analyzed was the ability of a subset of the kmer-SVM–predicted enhancers to direct expression of a luciferase reporter in vitro in melanocytes (n = 11). It was found that majority of enhancers tested direct luciferase expression in vitro more than threefold higher than the minimal promoter alone (8/11; 73%), and several drove expression more than fivefold (6/11; 55%) and 10-fold higher (3/11; 27%) (Fig. 7C). Also tested was the enhancer activity of three predicted enhancers in vivo using the same assay described above for ChIP-identified enhancers, and it was found that two of the three sequences assayed directed expression of GFP in the melanocytes of transgenic zebrafish (Table 6). GFP expression was mostly specific to melanocytes, though one predicted enhancer (no. 1) also directed expression in the CNS and otic vesicle. It should be noted that the predicted enhancers assayed here were chosen from among the predictions with the highest SVM scores rather than at random.

[201] To further demonstrate the power of this approach, genome-wide enhancer predictions in the human genome were made in the same way as described above for mouse. 7788 predicted melanocyte enhancers in the human genome were identified. Like the mouse predictions, the human predictions show strong sequence constraint (Fig. 7D), even though conservation was not taken into account when making predictions. The predicted human

enhancers display elevated levels of DNase I hypersensitivity (HS) in human primary melanocytes (data generated by The ENCODE Project Consortium) (Fig. 7E), which is a feature of active enhancers (Song and Crawford 2010; Song et al. 2011). Moreover, the degree of overlap between the kmer-SVM predictions and DNase I HS peaks was markedly higher in primary melanocytes and melanoma cell lines than in unrelated cell types (Fig. 7F), suggesting that the activity of the predicted enhancers is largely specific to the melanocyte lineage.

[202] The ability of the kmer-SVM classifier to make valid genome-wide predictions in the mouse and human genomes clearly demonstrates the high information content of the 6-mer vocabulary derived from the original training set. The kmer-SVM predictions also augment the catalog of putative melanocyte enhancers identified by adding an additional 7361 predicted enhancers in the mouse and 7788 in humans. Furthermore, the fact that a classifier trained on mouse sequences can make accurate predictions in the human genome clearly demonstrates the utility of this approach in identifying enhancers in genomes for which ChIP-seq data are not available, and provides direct proof of regulatory sequence vocabulary conserved between mouse and human.

DISCUSSION

[203] In this study, an approach to the investigation of regulatory sequences that integrates ChIP-based enhancer discovery with computational interrogation of sequence composition was demonstrated. The comprehensive nature of this approach represents a significant step forward in the ability to decipher the sequence basis of regulatory control of gene expression. Importantly, this strategy can be applied to any cell type of interest for which ChIP-seq and functional validation are feasible. This study began by employing ChIP-seq for EP300 and H3K4me1 to discover a large set of previously unidentified putative melanocyte enhancers. In the melan-a ChIP-seq data, a striking relationship between EP300 and H3K4me1 was observed, similar to that observed in other cell types (The ENCODE Project Consortium 2007; Heintzman et al. 2007, 2009; Ghisletti et al. 2010). The bimodal pattern of H3K4me1 ChIP-seq signal around EP300 peaks likely reflects the tendency of enhancers to be nucleosome depleted (Boyle et al. 2008; Song et al. 2011), and thus the flanking H3K4me1 signal arises from positioned nucleosomes marked by H3K4me1 on either side of the enhancer. A similar phenomenon was elegantly demonstrated in the case of nucleosomes at androgen-responsive enhancers in pancreatic cancer cells by He et al. (2010).

[204] Though other studies have employed ChIP-seq for EP300 alone to identify putative enhancers with notable success (Visel et al. 2009a; Blow et al. 2010), it was chosen to focus specifically on EP300 peaks flanked by H3K4me1 peaks as this approach minimized the inclusion of nonenhancer sequence features with the potential to obscure the sequence vocabulary underlying enhancer function. Though not the primary focus of this study, it was shown that there are significant differences between the subset of EP300 peaks that are flanked by H3K4me1 and those that are not and that these differences are consistent across unrelated cell types. These differences suggested that there is considerable value in using both EP300 and H3K4me1 data sets together for enhancer discovery, and that future studies to further unravel the relationship between EP300 and H3K4me1 are likely to yield important insights into enhancer biology.

[205] The rates of functional validation observed (86% in vitro and 70% in vivo) were consistent with validation rates of ChIP-seq identified enhancers reported previously, though there is considerable variation between studies (Heintzman et al. 2009; Visel et al. 2009a; Blow et al. 2010; Ghisletti et al. 2010). There was general agreement in activity between the in vitro Putative enhancers direct reporter expression in melanocytes of transgenic zebrafish embryos. Representative images for all seven enhancers positive in this assay showed GFP-positive melanocytes in transgenic (mosaic) embryos at 3 dpf after treatment with epinephrine. Six of seven elements showing activity in vivo also showed activity in vitro (threefold threshold). In addition, the enhancer with the strongest activity in vitro clearly had the strongest activity in vivo as well, as judged by the level of fluorescence in GFP-expressing melanocytes, the number of positive embryos observed, and the number of positive melanocytes per positive embryo. However, putative enhancer 3 drove melanocyte expression in vivo even though its enhancer activity was not significant in vitro, and conversely, three enhancers that drove expression in vitro did not drive expression in vivo in mosaic transgenic zebrafish (nos. 20, 25, and 30). These discrepancies between the results of the in vitro and in vivo functional assays used here could be the result of differences among the model organisms (mouse and zebrafish, respectively), the minimal promoters in the reporter constructs (E1B and FOS, respectively), or other limitations of the respective reporter assays. These results demonstrated the importance of using multiple complimentary assays to assess the function of putative enhancers.

[206] The orientation of the amplicon tested relative to the minimal promoter had a dramatic impact on the enhancer activity of sequences assayed *in vitro*. This was not likely to reflect an orientation dependence of the enhancer in its native genomic context. Rather, it was likely an artifact of the placement of the sequence in the synthetic context of a reporter construct. The orientation effect likely arose from the fact that the distance between an enhancer and minimal promoter in a reporter construct can strongly influence its functional output. This distance effect can be observed with as little 50 bp separating the two components (Nolis et al. 2009) and can manifest as orientation-dependent activity when testing an amplicon in which the critical sequence components (TF binding sites) are skewed to one side. In such a case, a given amplicon will show higher activity in the orientation that places its critical components closest to the minimal promoter, and lower (in some cases even undetectable) activity in the orientation that places its critical components furthest from the minimal promoter. Indeed, the strongest putative enhancer (no. 22), which mediates an increase of >100-fold reporter expression in the “forward” orientation and drives strong melanocyte expression *in vivo*, does not drive detectable expression *in vitro* in the “reverse” orientation (Table 5).

[207] The similarity between the motifs identified by DREME (Fig. 39C) and the 6-mers identified by the kmer-SVM classifier was strong evidence that these sequences are binding motifs for TFs that play significant roles in melanocyte biology. The identification of motifs predicted to bind SOX10 and MITF is consistent with the well-characterized roles for these TFs in the melanocyte line- age. JUN and FOS are major effectors of the MAP kinase signaling cascade, which is critical to the proliferation of melanocyte cells in culture (Swope et al. 1995). In addition, constitutive activation of the MAP kinase pathway is a hallmark of melanoma (Dutton- Regester and Hayward 2012). The enrichment for a motif predicted to bind members of the TEAD family may reflect an as yet unappreciated role for TEAD TFs in melanocytes. It does not appear that any TEAD family member has been previously shown to play a specific biological role in melanocytes. However, TEAD2 has been shown to bind an enhancer active in neural crest, the developmental precursor to melanocytes (Degenhardt et al. 2010). This binding causes an increase in the expression of Pax3, itself a TF that is predicted to bind one of the most highly weighted 6-mers.

[208] Motifs predicted to bind other TFs involved in melanocyte biology could have escaped detection due to high variation in consensus sequence, low enrichment relative to

negative control sequences, or inherent biases in the algorithms used here for motif detection. Additionally, the EP300/H3K4me1-based approach likely identified only a subset of enhancers active in melanocytes. This particular subset of enhancers may be more highly enriched for some TF binding sites than for others. Mechanistically distinct subsets of enhancers have been reported in other cell types (He et al. 2011a). Though beyond the scope of this study, ChIP-seq for additional factors and in additional melanocyte-related cellular substrates would likely help to distinguish potential differences between subsets of enhancers.

[209] Taken collectively, the melanocyte enhancers and corresponding sequence vocabulary described here greatly enhance understanding of the regulation of gene expression in melanocytes. Furthermore, they were relevant to human phenotypes and disease risk caused by variation in regulatory sequences. To date, at least 18 distinct genome-wide association studies (GWAS) have identified 52 SNPs associated with melanocyte-related phenotypes, including skin color, hair color, freckling, tanning response, number of cutaneous nevi, melanoma risk, and vitiligo (Hindorff et al. 2011). Many of these associations are likely to reflect causative variants that impact regulatory sequences (Hindorff et al. 2009; Visel et al. 2009b). This study, and others like it, promises to aid the identification of causative variants underlying genome-wide associations, as well as the molecular mechanisms by which they act.

METHODS

ChIP-seq

[210] Melan-a cells were propagated according to guidelines from Sviderskaya et al. (2002). ChIP was performed according to the method previously described (Lee et al. 2006). Alternative lysis buffers to those in the referenced protocol were used as follows: lysis buffer 1 (5 mM PIPES, 85 mM KCl, 0.5% NP-40, and 1× Roche Complete, EDTA-free protease inhibitor), lysis buffer 2 (50 mM Tris-HCl, 10 mM EDTA, 1% SDS, and 1× Roche Complete, EDTA-free protease inhibitor), and lysis buffer 3 (16.7 mM Tris-HCl, 1.2 mM EDTA, 167 mM NaCl, 0.01% SDS, 1.1% Triton X-100, and 1× Roche Complete, EDTA-free protease inhibitor). Sonication was performed using a Bioruptor (Diagenode) with the following settings: high output; 30-sec disruption; 30-sec cooling; total sonication time of 35 min with addition of fresh ice and cold water to water bath every 10 min. Four micrograms of ab8895 (Abcam) and 10 mg of antibody sc-585 (Santa Cruz Biotechnology) were used for H3K4me1

and EP300 ChIP, respectively. IP wash conditions were adjusted from the protocol referenced above as follows: Each immunoprecipitation (IP) was washed twice with low-salt wash buffer (0.1% SDS, 1% Triton X-100, 2 mM EDTA, 20 mM Tris-HCl, 150 mM NaCl), twice with high-salt wash buffer (0.1% SDS, 1% Triton X-100, 2 mM EDTA, 20 mM Tris-HCl, 500 mM NaCl), and twice with LiCl wash buffer (0.25 M LiCl, 1% IGEPAL CA630, 1% deoxycholic acid [sodium salt], 1 mM EDTA, 10 mM Tris-HCl) and rinsed once with PBS (pH 7.4). At least two biological replicates were performed for each antibody, with each replicate consisting of a ChIP sample and an input (pre-IP) sample. Each replicate was performed with $\sim 1 \times 10^8$ melan-a cells. ChIP libraries were submitted to NIH Intramural Sequencing Center, and each was sequenced on one lane of an Illumina GA2 yielding >20 million reads per sample, with the exception that each EP300 ChIP library was sequenced on two lanes for increased coverage depth.

Analysis of ChIP-seq data: peak calling

[211] EP300 peaks were called using the Model-based Analysis for ChIP-seq (MACS) algorithm (Zhang et al. 2008). Peaks were called for each replicate independently, and only those that were called in both replicates ($n = 3622$) were selected for further analysis. Coordinates reported are from Replicate 1. H3K4me1 peaks were called using cisGenome (Ji et al. 2008) because it tends to call separate peaks corresponding to each apex of the bimodal distribution of H3K4me1 signal flanking enhancers, whereas MACS tends to call the entire bimodal distribution as a single peak. The Two Sample Peak Calling option in cisGenome was used, which allows both replicates to be entered simultaneously to produce a single set of output files. Default settings were used for both peak callers, except that 'half window size W' was set to 4 for cisGenome.

Distribution of ChIP-seq reads relative to features of interest

[212] The total number of sequencing reads covering each base in a window of indicated size (x-axis) around the summit/center of the set of genome regions of interest (ChIP-seq peaks/kmer-SVM predictions) was calculated with a custom script. The total number of reads covering each base in the window was then smoothed in 100 bp bins, and is represented as 'reads' (y-axis) in Figures 38C. For Figure 41B and Figure 16, a subsequent calculation was performed in which the total reads in each bin was divided by the number of genome regions in the set of interest, to facilitate comparison between sets of different sizes. This

normalized measure is represented as “Avg reads per peak” (y-axis) in Figure 41B and Figure 16. The heatmap in Figure 39D was generated with the heatmap tool in the Cistrome Analysis Pipeline (Liu et al. 2011) using a bed file of 3622 EP300 peaks (300-bp regions centered the peak summits), and a wig file of H3K4me1 ChIP enrichment generated by MACS as standard output from peak calling.

ENCODE data

[213] ENCODE data in Figure 41 was processed as described above for melan-a data. Much of the data handling for these analyses was performed with Galaxy (Giardine et al. 2005; Blankenberg et al. 2010; Goecks et al. 2010).

In silico analysis of putative enhancers:

Average phastCons score

[214] Average phastCons score plots (Figs. 39B) were generated with the Conservation Plot tool as part of the Cistrome Analysis Pipeline using an interval file of H3K4me1-flanked EP300 peaks (300-bp intervals around peak summits) (Fig. 39B) or kmer-SVM predicted enhancers.

Motif analysis

[215] DREME (Bailey 2011) was used to identify enriched motifs (Fig. 2C). Sequences of 2489 putative melanocyte enhancers (centered on the EP300 ChIP-seq peak summit and extending ± 150 bp) were used as input. Default settings for motif size (mink = 3, maxk = 7) were used. Motifs were submitted to TOMTOM (Gupta et al. 2007) as part of the MEME Suite (Bailey et al. 2009) to predict binding factors corresponding to each enriched motif, and the top vertebrate TF match was reported unless otherwise indicated in text. In the case of MITF and PAX3 (Figs. 39C; Fig. 15), match was made based on high similarity to published binding specificities (Bentley et al. 1994; Chalepakakis and Gruss 1995; Yasumoto et al. 1995), as there is no position weight matrix (PWM) for either of these TFs in the databases queried by TOMTOM (JASPER and UniProbe).

GO analysis

[216] GREAT (McLean et al. 2010) was used to identify GO terms enriched among genes

proximal to putative enhancers. The association rule was set as follows: proximal, 50 kb upstream and 50 kb downstream (any gene in this interval relative to input regions is included); plus distal, up to 500 kb (if no gene is present in the proximal interval, the closest gene in this distal interval is included). For details, see McLean et al. (2010).

Distribution of enhancers relative to genes expressed at different levels in melan-a

[217] Previously published melan-a microarray data were used (Buac et al. 2009). For analyses in Figure 39, only genes represented on the array with a corresponding TSS in RefSeq (n = 17,957) were used. These genes were ranked by raw expression level in melan-a (probes averaged, mean of three replicates). Custom scripts were used to calculate the number of putative enhancers within 500 kb (in bins of 100 kb) (Fig. 39D) and 5 kb (in bins of 1 kb) (Fig. 39E) of TSSs of the top 2000 and bottom 2000 genes on the ranked list, as well as for five sets of 2000 genes selected randomly from this list.

Luciferase assays

[218] All tested sequences (putative enhancers, negative regions, kmer-SVM predictions, and previously characterized enhancers) were PCR amplified from mouse genomic DNA (Promega, no. G309A) and TA-cloned with the pCR8/GW/TOPO TA Cloning kit (Life Technologies). The luciferase reporter construct contains the firefly luciferase gene downstream from a minimal E1B promoter (Anto-nellis et al. 2006). Test sequences were inserted into a gateway cloning site upstream of the promoter with a directional LR reaction (Gateway cloning from Life Technologies). All sequences were tested in both orientations, and data from the orientation with the highest expression were used for downstream analysis to give the most accurate representation of the potential of each sequence to drive expression in melanocytes. For negative control regions, a set of 2000 regions was generated in which the regions were matched to the putative enhancers in size, GC%, and repeat fraction, but with a read count below for EP300 and H3K4me1. Ten regions were selected at random from this set for functional testing. For all luciferase assays, melan-a cells were plated in 24-well format (40,000 cells/well) and transfected next day with 400 ng of luciferase reporter and 8 ng of pCMV-RL Renilla expression vector (Promega) using 2 mL Lipofectamine 2000 per well (Life Technologies). Cell lysate was collected at 48 h post-transfection and assayed with the Dual-Luciferase Reporter Assay System (Promega) using a Tecan GENiosPro Microplate Reader (Tecan Group). Three biological replicates were performed for each construct.

Zebrafish transgenesis

[219] All tested sequences were PCR amplified and TA-cloned as described above (see Luciferase Assays). The GFP reporter construct, described previously (Fisher et al. 2006b), contains a gateway recombination cassette (Life Technologies) upstream of a minimal (FOS) promoter and EGFP. The reporter used here was modified slightly by insertion of an eye-specific regulatory element from the zebrafish *crybb1* locus (chr10:45,529,501–45,530,122; Zv9) downstream from EGFP to facilitate screening for successful transgenesis independent of the test sequence. Zebrafish transgenesis was performed as previously described (Fisher et al. 2006b). Briefly, each construct was injected into >150 wild-type (AB) embryos at the one- to two-cell stage with Tol2 transposase mRNA to facilitate efficient and random integration of the reporter construct (flanked by tol2 recombination arms) into the zebrafish genome. Embryos were screened for GFP expression at 3 d post-fertilization (dpf), a timepoint at which melanocytes are well developed and the embryos are most amenable to comprehensive screening. Embryos were also screened at 2, 4, and 5 dpf, albeit less thoroughly, and no significant differences in expression from 3 dpf were observed. At least 10 positive embryos were imaged at 3 dpf for each positive construct. For high-magnification fluorescent images of melanocytes, zebrafish were treated with epinephrine 5–10 min prior to imaging (4 mg/mL) in order to contract pigment granules toward the center of the cell and thus facilitate visualization of GFP at the periphery. For full-body lateral images embryos were raised in 1-phenyl 2-thiourea (PTU) from 24 hpf until imaging to inhibit melanin synthesis. All Images were taken on a Nikon AZ100 Multizoom microscope with NIS-elements software. All zebrafish work was performed under an approved protocol (FI10M369), reviewed by the Johns Hopkins Institutional Animal Care and Use Committee.

Kmer-SVM classifier

[220] To generate a high-confidence training set, a new set of 400-bp regions was defined that maximizes the overall EP300 ChIP-seq signal within each of the 2489 putative melanocyte enhancers after removing any enhancers which were >70% repeats. Repeat masked sequence data (mm9) was used from the UCSC Genome Browser to calculate repeat fractions. For negative sequences, a 50× larger set of random genomic 400-bp sequences were found by matching GC and repeat fraction of the positive set. Additionally, any potential EP300-bound regions with Poisson test P-value <0.1 (10 ChIP-seq reads) were excluded. At each sampling step, a region from the positive set was randomly selected, the

GC content and the repeat fraction were calculated, a genomic sequence that matched these properties was sampled, and sampling was repeated until obtained 50× sequences were obtained. Standard fivefold cross validation was performed to assess the performance of this kmer-SVM classifier. The quality of the classifier was measured by calculating the auROC, which plots the true positive rate vs. the false-positive rate of the predictions. The PRC is a more reliable measure of performance than the ROC when positive and negative sets are unbalanced, as in this case. Precision is the ratio of true positives to predicted positives, and recall is identical to the true positive rate in the ROC. The PRCs can be quantified by the auPRC, or average precision. TFs predicted to bind top 6-mers were determined as described above for DREME motifs (see Motif Analysis). Predictions for functional validation (n = 11) were chosen from the top of a list of regions ranked by SVM score. These are not the top 11 ranked predictions overall however, because the list they were chosen from was generated by an earlier version of the classifier trained on a slightly different input set. In the final set of predictions, the 11 regions tested are ranked by SVM score as 13, 15, 1, 9, 2, 44, 21, 108, 24, 273, and 203, respectively.

EXAMPLE 4

Prediction of estrogen-related-receptor beta bound regions in mouse ES cells

[221] To take a specific example, the ChIP-seq data set of Chen et al. (2008), who identified binding loci of TFs in mouse embryonic stem (ES) cells was first considered. As an example, their ChIP-seq data was analyzed for estrogen-related-receptor beta (ESRRB) known to play a role in maintaining the pluripotency of ES cells. Because the ESRRB bound regions reported by Chen et al. (2008) were short (10–30 bp), we extended from the midpoint of these regions and used 100 bp elements as the positive sequence set. Following the workflow in Figure 32, then a 10× negative set was used to train the SVM, then generated the ROC and PR curves for Chen's ESRRB data set as shown in Figure 33A. These curves are typical of an accurate classifier, and summary statistics of AUROC = 0.921 and AUPRC = 0.74 for this data set were obtained. To directly compare the kmer-SVM prediction results with the PWM scores, the maximum log-odd score of the ESRRB PWM was calculated for each sequence and then plotted the ROC and PR curves as shown in Figure 33B. Although the ESRRB PWM is regarded as an easy motif, its classification performance (AUROC = 0.88 and AUPRC=0.654) is significantly lower than kmer-SVM.

[222] The top five positive and negative kmers reported by the trained SVM were shown in Figure 33C. Also in Figure 33C for comparison was the PWM for ESRRB found and reported in Chen et al. (2008). As expected, the top kmers span the core motif of the ESRRB-binding site, but interestingly, several SVM-predicted kmers contribute to the specificity of the ESRRB. For example, AAGGTC (first), AGGTCA (second), CAAGGT (third), AGGTC G (fourth) and so forth have large positive weights, but A GGTCC and AGGTCT have large negative weights, showing that A or G is allowed in the binding site at the 11th position of the PWM, but that C and T are not. This subtlety is not reflected in the PWM found by Weeder, the motif discovery algorithm used in Chen et al. (2008).

Prediction of distinct Glucocorticoid receptor bound regions in 3134 and AtT20 cells

[223] Next it was shown how a kmer-SVM can be applied to identify sequence features responsible for directing the binding of a single TF to different genomic locations in distinct tissues, developmental states or cell lines. As an example, John et al. (30) investigated the genomic binding of the Glucocorticoid Receptor (GR) TF in response to hormone stimulation in two divergent cell lines. Specifically, GR binding was profiled via ChIP-seq on a mouse mammary adenocarcinoma derived cell line (3134) and mouse pituitary (AtT20) cells. The binding of GR in these two cell lines were largely at non-overlapping genomic loci. John et al. (30) showed that the consensus GR-binding element (GRBE) was present in both 3134 and AtT20 bound regions, but that distinct sets of accessory sequence motifs were detected in the two cell lines, including binding sites for AP1, AML1, HNF3, TAL1 and NF1.

[224] A method of the present disclosure was followed to train a kmer-SVM on the ChIP-seq GR bound loci in 3134 cells versus 10× random genomic sequence and separately on GR bound loci in AtT20 cells versus 10× random genomic sequence, using the coordinates in John et al. (30) as positive set input. This kmer-SVM classifier achieved an AUROC of 0.901 and AUPRC of 0.569 in 3134 cells, and AUROC of 0.909 and AUPRC of 0.596 in the AtT20 cell line (Figure 34A), indicating that GR binding in both cell lines is predictable based on sequence. The top 10 positive and negative weight kmers for each cell line are shown in Figure 34A, recovering kmers that span the GRBE and binding sites for accessory factors reported in John et al. (30). Although high scoring kmers matching the GRBE consensus were found in both cell lines, the accessory factors are specific to each cell line. In 3134 cells, the top two ranking kmers both match AP-1, and the eighth and ninth highest kmers in 3134 cells matched AML1. The kmer-SVM also identified TEAD1 as the fifth most important

kmer (ACATTC), a binding site not found in John et al. (30). In addition, four of the most negative kmers match the binding site for ZEB1 or Snail, a common negative sequence feature in the analysis, indicating that the absence of ACCT or AGGT is predictive for GR bound regions. Thus, it was hypothesized that either the presence of a ZEB1-binding site would directly inhibit the binding of GR, presumably through the binding of ZEB1 or another factor that binds specifically to this site. In other cases, this binding site could otherwise disrupt the normal function of the enhancer elements and is thus required to be absent (3).

[225] In the AtT20 cells, a separate set of accessory sites was found: the fourth, fifth and seventh most positive kmers match HNF3, whereas the second and third match TAL1. The sixth ranked kmer matched NF1. The eighth and tenth ranked kmers match CREB, not reported in John et al. (30). In summary, this analysis uncovered most of the accessory factors described in John et al. (30), but also identifies novel positive and novel negative binding sites. Further, it is demonstrated that these features are predictive, in the sense that these features can be used to accurately classify the positive and negative regions, and were not simply over (or under) represented in one of the sets.

[226] Next, it was demonstrated that the kmer-SVM is able to directly distinguish the GR bound regions in 3134 cells from the GR-bound regions in AtT20 cells from DNA sequence. In this case, random genomic sequence were not used as the negative set, but instead a kmer-SVM was trained using the AtT20 regions as the positive sequence set, and the 3134 regions as the negative sequence set. The ROC and PR curves are shown in Figure 35A, yielding AUROC of 0.889 and AUPRC of 0.794. Thus, DNA sequence is sufficient to distinguish the cell specific binding of GR. Now, as both sets are bound by GR, the kmer weights shown in Figure 35A do not include the GRBE, as it is present in both sets. The distinguishing features are now binding sites for the GR accessory factors. The kmer CAGGTG (ZEB1), which was negative for 3134 versus random is now the most positive kmer for AtT20 versus 3134. The other positive kmers match the AtT20-specific accessory factors TAL1 and HNF3. The negative weight kmers are the 3134 specific accessory factors AML1 and AP1. This demonstrates that these accessory sequence elements are predictive of the tissue-specific binding of GR because the sequence information in the accessory factor-binding sites is sufficient to distinguish GR binding in these two contexts. It is emphasized that this is a stronger statement than simply observing the enrichment of distinct sequence features in the two cases: A further hypothesis is proposed that these sequence features are sufficient to

specify which GR-binding sites will be occupied in each tissue. This differential occupancy is determined by the presence of binding sites for accessory factors, which can be identified from the kmer weights.

EXAMPLE 5

Prediction of distinct EWS-FLI bound regions in EWS502 and HUVEC cells

[227] Although the previous example showed that binding of a sequence specific TF to different loci in different tissues was predictable from DNA sequence, now turn to an example where a wild-type and mutant TF were shown to bind distinct regions, and that this differential binding is also predictable from DNA sequence. Most Ewing- Sarcoma tumors harbor a mutation, which creates an oncogenic chimerical EWS-FLI TF by fusing the transactivation domain of EWS to the DNA-binding domain of FLI. Patel et al. (55) showed that this chimeric EWS-FLI TF targets different genomic regions in tumor cells and in non-tumor cells, and that additionally the wild-type protein FLI1 binds to largely the same regions as the fusion protein in non-tumor cells. Specifically, the authors assayed binding in the EWS502 cell line (derived from a Ewing Sarcoma tumor) and primary human endothelial cells (HUVEC). They reported a preferential binding for regions containing repeats of the tetranucleotide GGAA by EWS-FLI in both EWS502 and HUVEC cells (although the tumor cell line showed a greater enrichment). Additionally, binding of EWS-FLI in HUVEC cells was shown to be enriched in ETS, AP1 and GATA motifs, but that these accessory motifs were largely absent from the EWS-FLI bound regions in EWS502 cells.

[228] To analyze these data sets, used as positive sets were the ChIP-seq regions in Patel et al. (55) bound by EWS-FLI in EWS502 cells and HUVEC cells, and separate 10x negative sets were generated for each cell line. After training the kmer-SVM, in EWS502 cells, the AUROC was 0.965 and AUPRC was 0.884, and in HUVEC cells, the AUROC for this data set was 0.964 and AUPRC was 0.798 (Figure 36A), again showing that the cell line specific binding of the EWS-FLI TF is predictable from primary DNA sequence features. In this case, the training data were optimized for length by the peak-calling algorithm ZINBA (28), which may account for the extremely high classification performance. Another possible factor is that the repeat fraction in these positive sets is relatively high.

[229] Our method finds some motifs common to both cell lines. Positive sequence features reflect both the ETS motif recognized by FLI1 and the repetitive structure reported by Patel

et al., with the ETS motif GGAA as part of the highest ranked kmers in both cell lines, as shown in Figure 36B. Negative weight kmers are again found to be significant. Kmers that disrupt the repetitive GGAA structure (e.g. TGGAAG) score negatively in both cell lines, but more negatively in EWS502 cells. Notably, many of the most negative kmers for both cell lines contain AGGT, again emphasizing the importance of the absence of ZEB1 or Snail repressor family-binding sites for EWS-FLI binding or function.

[230] Cell line-specific kmers recover the AP1 motif reported in Patel et al. (55), and a potentially novel role for TEAD1. The HUVEC specific accessory factor AP1 is found as a high scoring motif in HUVEC cells, but not EWS502 cells. Two highly negative kmers in EWS502 cells correspond to the binding site for TEAD1. TEAD1 has been implicated in tumor suppression and growth control and because the absence of TEAD1 binding sites is predictive of EWS-FLI binding in EWS502 cells, but not HUVEC cells, it is tempting to speculate that TEAD1-binding would disrupt EWS-FLI binding in EWS502 cells, but not in HUVEC cells.

EXAMPLE 6

Kmer-SVM versus PWM

[231] To systematically evaluate the kmer-SVM method on a more exhaustive collection of data, all CHIP-seq data sets generated as part of the ENCODE project (29,30) were analyzed. The 467 sets of peaks generated by ENCODE Uniform processing pipeline (29), after removing any data sets containing <500 peaks (27 sets were excluded by this criterion) were used. Then a kmer-SVM model was trained on each set versus an equal size (1x) set of corresponding random genomic regions and calculated the AUROC. As a comparison, the AUROC of each single PWM was independently calculated in a combined database of 890 PWMs, using as predictors the PWM score of the top hit in each region. Figure 37 shows that the kmer-SVM prediction outperforms the best single PWM in almost all cases. The only notable exception is the CTCF PWM (red circles), which is predictive for CHIP on CTCF and members of the cohesin complex (RAD21, SMC3), which are known to co-localize with CTCF. CTCF is one of the longest and information rich PWMs and seems to operate in a non-combinatorial manner; therefore, it seemed to be relatively unique in that its genomic binding can be predicted with a single PWM. In addition, its long binding site was not handled optimally by the current kmer-SVM model.

DISCUSSION

[232] It is shown that a kmer-SVM model as offered via a web server was able to find predictive sets of DNA sequence features in several different genomic data sets and can be used to assess and explore the genomic data and generate testable hypotheses for subsequent biological analysis. Using the existing sequence tools and pipeline flow of the Galaxy platform has greatly facilitated the ease of distribution. The examples, in addition to the previous results on mouse EP300 bound enhancers and melanocyte enhancers, emphasized several key benefits of the kmer-SVM analysis. Using a web server, users can find the essential sequence features, which distinguish a set of experimentally determined genomic regions from random sequence, and identify key accessory factors and repressive elements for biological interpretation and follow-up investigations. In addition, users can use the kmer-SVM to score alternative sequence sets or entire genomes to make predictions of the activity of these regions in the relevant context.

[233] A web server may provide complementarity to existing PWM discovery and scoring tools, including XXmotif, MEME, SCOPE, RSAT, RegAnalyst and Amadeus. XXmotif operates by attempting to optimize the statistical significance of a given PWM. Specifically, XXmotif develops and then iteratively merges PWMs for motifs until P-values cannot be improved. The core of MEME is the use of mixture models, arrived at by means of expectation maximization, to identify motifs. SCOPE uses three different algorithms, separately directed toward identify short non-degenerate motifs, short degenerate motifs and long degenerate motifs and uses a scoring method to integrate the output from each of these algorithms. SCOPE is a parameter-free program and requires no parameters to be provided by the user. RSAT is a more general toolbox for the analysis of sequence data and uses a tool for motif discovery, which compares the observed occurrence of motifs against the expected presence of that motif, given the distribution of nucleotide occurrence in an organism (37). RegAnalyst uses a series of thresholds applied to the counts of motifs observed in a set of sequences. Amadeus also compares the frequency of the presence of motifs against a background model. In contrast, the web server SVM method shown herein focused on finding combinations of sequence features, which are usually more predictive than single motifs, as show in Figure 37.

[234] As is currently known, there is only one web server available (<http://galaxy.raetschlab.org/>) that offers simple SVM functions including several string kernels as

well as other common kernels, such as linear and Gaussian. It also provides means to evaluate prediction performance using ROC and PR curves. This server, however, is mainly intended for general use of SVMs by users with a certain level of computational experience. In contrast, the kmer-SVM web method disclosed herein was designed to allow biologists with no prior machine learning expertise to quickly and rigorously analyze regulatory sequence data sets. To do so, methods herein incorporated steps with functionality required for regulatory sequence analyses and took into account the specific properties of regulatory elements. First, the spectrum kernel function was modified to account for the fact that TFs bind to double-stranded DNA. Not only was an exact kmer counted but also counted was its reverse complement kmer. Redundant kmers were then eliminated from the final feature set to remove the possible bias caused by double counting. Second, a step that generated negative sequence sets to match the distribution of sequence length, GC content and repeat fraction of the corresponding positive sets was used. This ensured that the SVM classification reflects the most biologically relevant mechanisms. Third, provided was a means to interpret and explain the results by calculating the SVM weights of kmers from a list of support vectors, the primary output of SVM training. None of these functionalities provided by the present disclosure provided on a web server is available at the Galaxy server at Ratsch's laboratory.

EXAMPLE 7 Gapped k-mers

[235] k-mer based approaches may have difficulty in estimating long k-mer frequencies in a finite set of biological samples. Presented herein is a general solution to this problem, and the method can be applied to improve the statistical robustness of any of the aforementioned k-mer based approaches or others which use k-mer frequencies as direct features or as an intermediate step in the construction of more complex sequence descriptors.

[236] When using k-mers, larger k's will resolve larger binding sites and more accurately reflect biological function. For example, some transcription factors (such as ABF1 or CTCF) have relatively long binding sites that cannot be completely represented by short k-mers. So longer k-mers capture more relevant information; however, there is a limitation on the maximum length k which can be effectively used in statistical algorithms. Because longer k-mers are more sparsely populated in any finite training sequence set, there is a maximum length k for which the k-mer frequencies can be robustly estimated. Thus in practice, a k is chosen which is a tradeoff between resolving features and robust estimation of their frequencies. To overcome the finite training set size problem, the present disclosure may

employ gapped k-mer frequencies. A gapped k-mer has a length l , and a number of informative columns within that l -mer, k , which reflects the base pairs which actually affect the strength of the TF-DNA binding interaction. It was found that using gapped k-mers may improve the reliability of the l -mer frequency estimation for a finite genomic training set, because while l -mers become sparsely populated, gapped k -mers will still have many instances in the training set, and thus their frequencies can be more reliably estimated. The observed gapped k-mer frequency distribution was used for all gapped k-mers to estimate the ungapped l -mer frequencies, which are sparsely populated. Mathematically this turned out to be the minimum norm estimate for the l -mer frequencies given the frequencies for all gapped k-mers. The matrix, W , mapping between these two spaces was derived. A closed form for this matrix was obtained by studying the combinatorial properties of the incidence matrix.

Problem statement

[237] In any given sequence sample, there are observed counts of gapped k-mers and ungapped l -mers. The fundamental assumption is that the counts of gapped k-mers in this sample are sufficient to define its biological function, and are also more robustly estimated from the sequence sample. Instead of using the observed counts of ungapped l -mers, the most robust set of ungapped l -mers were sought that was also consistent with the gapped k-mer distribution. By robust, it is meant that this estimate is resilient to small changes in the input or training set, even for large l for which the actual l -mer counts are very sparse. A classifier based on these more robustly estimated l -mer counts was consequently also more robust in the sense that it was less sensitive to small changes in the input or training set, and is therefore a more stable predictor than a classifier based on exact counts.

[238] Because the mapping from ungapped l -mers to gapped k-mers is underdetermined, there are many sets of l -mer counts that could have produced the observed set of gapped k-mer counts. Proposed as the best set of l -mer counts is the minimum norm l -mer distribution consistent with the gapped k-mer distribution. This is also the solution that minimizes the mean-squared error from a constant (flat) l -mer distribution. The observed set of ungapped l -mers is only one set of sequences which would have produced the given gapped k-mer counts. The minimum norm distribution was in some sense the most likely of all sequences which could have produced the observed gapped k-mer counts.

[239] For the case of DNA sequences, the alphabet is $\{A, C, G, T\}$, so the length of the alphabet is $b = 4$, but the solution for the optimal ℓ -mer count distribution presented below is valid for any b . Ungapped ℓ -mers, and gapped k -mers of length ℓ with k ungapped (informative) positions were considered.

Definition 1 The set of ungapped ℓ -mers is $U = \{u_j\}$, $1 \leq j \leq N = b^\ell$, the set of all different sequences of length ℓ over the alphabet $\{0, 1, \dots, b-1\}$.

Definition 2 The set of gapped k -mers is $V = \{v_i\}$, $1 \leq i \leq M = \binom{\ell}{k} b^k$, the set of all gapped k -mers of length ℓ with k known bits and $\ell - k$ gaps, where $k < \ell$.

Definition 3 The matrix which maps ungapped ℓ -mers to gapped k -mers is the matrix $A^{M \times N} = [a_{i,j}]$, a binary matrix defined as the following:

$$a_{i,j} = \begin{cases} 1 & \text{if } v_i \text{ matches } u_j \\ 0 & \text{otherwise} \end{cases}$$

Here “ v_i matches u_j ” means that all ungapped positions in the gapped k -mer v_i have the same letter of the alphabet as the corresponding position in the ungapped ℓ -mer u_j .

The ungapped count vector is defined as follows:

Definition 4 $\mathbf{x}$ is a vector of length N , where x_j is the count for u_j , and the gapped count vector is:

Definition 5 $\mathbf{y}$ is a vector of length M , where y_i is the count for v_i .

[240] Given the above definitions, the mapping from ℓ -mer counts to gapped k -mer counts can be written as:

$$\mathbf{y} = A\mathbf{x} \quad (1)$$

As shown below, while it is usually but not always the case that $M < N$, since the rank of the matrix A , $\text{rank}(A) = \sum_{s=0}^k \binom{\ell}{s} (b-1)^s < N = b^\ell = \sum_{s=0}^{\ell} \binom{\ell}{s} (b-1)^s$ for $k < \ell$, this system is always underdetermined. Therefore, there are many possible ℓ -mer count vectors $\mathbf{x}$ that would produce the same gapped k -mer count vector $\mathbf{y}$. While the maximum entropy $\mathbf{x}$ is probably the most robust estimate to use, its solution is nonlinear and would likely require

prohibitive numerical computation. As a reasonable and tractable alternative, chosen as the next best alternative, was the minimum L2-norm solution to Eq. (1), $\hat{\mathbf{x}}$.

Theorem 1 Suppose that the matrices A , Q and A are defined as above. Then the minimum norm estimate for $\mathbf{x}$ is given by $\hat{\mathbf{x}} = W\mathbf{y}$, where $W_{N \times M}$ can be written as the following:

$$W = A^T Q \Lambda^{-1} Q^T \quad (2)$$

Proof To find the $\mathbf{x}$ which minimizes the L2-norm $\mathbf{x}^T \mathbf{x}$, let

$$F = \frac{1}{2} \mathbf{x}^T \mathbf{x} - \lambda \mathbf{f}, \quad (3)$$

where $\mathbf{f} = A\mathbf{x} - \mathbf{y} = 0$ is the constraint that satisfies (1), and λ is a vector of Lagrange multipliers. Minimizing yields

$$\frac{\partial F}{\partial \mathbf{x}} = \mathbf{x} - A^T \lambda = 0 \quad \text{and} \quad \frac{\partial F}{\partial \lambda} = -(A\mathbf{x} - \mathbf{y}) = 0. \quad (4)$$

Since A^T is not invertible, solve for $\mathbf{x}$ from $\mathbf{x} = A^T \lambda$ and use this in $A\mathbf{x} - \mathbf{y} = 0$ to get

$$\mathbf{y} = AA^T \lambda. \quad (5)$$

Since AA^T is a positive semidefinite matrix, it admits the eigendecomposition $AA^T = Q\Lambda Q^T$ where the matrix Λ is a diagonal matrix having nonzero eigenvalues of AA^T on its diagonal and the columns of Q are normalized orthogonal eigenvectors ordered similarly. It is obvious that $Q^T Q = I$ and it is not hard to prove that $QQ^T A = A$

Multiplying on the left by $A^T Q \Lambda^{-1} Q^T$ yields the minimum norm solution:

$$W\mathbf{y} = A^T Q \Lambda^{-1} Q^T \mathbf{y} = A^T \lambda = \hat{\mathbf{x}}. \quad (6)$$

[241] The derivation of a simple form for the matrix $W = A^T Q \Lambda^{-1} Q^T$, which generates the minimum norm $\mathbf{x}$ from the observed gapped k-mer counts $\mathbf{y}$, by $\hat{\mathbf{x}} = W\mathbf{y}$, is the main result of this paper.

[242] It is worth noting that the minimum L2-norm $\mathbf{x}$ can also be thought of as the minimum mean square error or the most likely distribution under the assumption that the x_j 's are independent and have a joint normal prior probability distribution with equal variances and expected values for all the ℓ -mer counts:

$$F(\mathbf{x}) = (2\pi)^{-\frac{N}{2}} |\Sigma_{\mathbf{x}}|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x}-\bar{\mathbf{x}})} \quad (7)$$

where $\Sigma_{\mathbf{x}} = \sigma^2 I_N$ is a diagonal matrix with constant elements on the diagonal and $\bar{\mathbf{x}}$ is a constant vector, i.e. $\bar{x}_i = c$. The $\mathbf{x}$ that maximizes (7) subject to the constraints given by (1) turns out to be the minimum norm solution. The proof for this is very similar to the proof of Theorem 1, as follows.

[243] *Proof* Applying the Lagrange multipliers technique to find the $\mathbf{x}$ that maximizes the logarithm of $F(\mathbf{x})$ subject to the constraints given in (1):

$$\mathbf{x} - \bar{\mathbf{x}} = A^T \lambda. \quad (8)$$

Reordering (8) and applying (1) the following was obtained

$$\mathbf{y} = AA^T \lambda + A\bar{\mathbf{x}}. \quad (9)$$

Now, consider the following eigendecomposition for AA^T :

$$AA^T = Q\Lambda Q^T. \quad (10)$$

Multiplying both sides of (9) by $A^T Q\Lambda^{-1} Q^T$ and applying (10) the following was obtained

$$\begin{aligned} A^T Q\Lambda^{-1} Q^T (\mathbf{y} - A\bar{\mathbf{x}}) &= A^T Q\Lambda^{-1} Q^T AA^T \lambda \\ &= A^T \lambda. \end{aligned} \quad (11)$$

Reordering (11) and applying (8) the following was obtained

$$\hat{\mathbf{x}} = W(\mathbf{y} - A\bar{\mathbf{x}}) + \bar{\mathbf{x}}. \quad (12)$$

[244] In the case disclosed herein, there is no difference between the expected values for different ℓ -mers counts, i.e. $\bar{x}_i = c$. Also it can be shown that the sum of the elements in rows of matrix WA is equal to 1, therefore, in this case $WA\bar{\mathbf{x}} = I\bar{\mathbf{x}}$, and equation (12) can be simplified to

$$\hat{\mathbf{x}} = W\mathbf{y} = A^T Q\Lambda^{-1} Q^T \mathbf{y}. \quad (13)$$

hence $W = A^T Q \Lambda^{-1} Q^T$, as required. Note that $\hat{\mathbf{x}}$ is independent of $\mathbf{x}$ only if all the c -mers have equal expected counts.

[245] The derivation of an explicit form for the matrix W , which is shown to be the Moore-Penrose pseudoinverse of A and maps gapped k -mer counts to ungapped ℓ -mers count estimates, is the central result of this Example. Although Eq. (2) gives a method to obtain the weight matrix W from the eigendecomposition of matrix AA^T given in (10), numerical calculation of the eigenvectors for AA^T would be computationally expensive as the size of matrix A grows very rapidly for large values of ℓ , k , and b . For example, for $(\ell = 15, k = 7, b = 4)$, there is $N \cong 10^9$, and $M \cong 10^8$. However, considering the symmetry of matrix A , shown in the detailed proof that follows is that the matrix W has a simple structure. In this matrix, the entry $w_{i,j}$ only depends on the number of mismatches between the ℓ -mers u_i and the gapped- k mer v_j . So there exists a finite sequence of only $k + 1$ values $w_0, w_1, \dots, w_k$ such that $w_{i,j} = w_m$ if u_i and v_j have exactly m mismatches. A *mismatch* is defined to be a difference between a gapped k -mer and an ℓ -mer in an ungapped position. So, for example, if N denotes a gap, $N A N G$ and $A C G G$ have one mismatch, and $A N G G$ and $A C G G$ have zero mismatches. w_m is largest for $m = 0$ and typically becomes negative for $m = 1$, and then oscillates about zero. See Sect. 7 for a concrete example. Thus, the entries of matrix W are limited to a small set of values $\{w_0, \dots, w_k\}$ and these values are specified by the following theorem:

[246] **Theorem 2** *The values of the elements of matrix W are given by the following equation, in which, ℓ is the sequence length, b is the size of the alphabet, k is the number of known bits, and m is the number of mismatches between the corresponding ℓ -mer u_i and the gapped- k mer v_j :*

$$w_{\ell,k,m} = \frac{\binom{k-\ell}{m}}{b^\ell \binom{\ell}{k} \binom{k}{m}} \sum_{n=0}^{k-m} \binom{\ell}{n} (b-1)^n \quad (14)$$

[247] Note that matrix W clearly depends on ℓ , k and b but for fixed ℓ , k and b , the entry on row i and column j of this matrix only depends on m , the number of mismatches between v_i and u_j , i.e. differences between ungapped positions in the gapped k -mer and the ungapped ℓ -

mer. Hence elements of W are limited to a small set of $k + 1$ values, as specified by the above theorem, and is very simple and easy to compute.

Example 8 Gapped K-mers for Enhanced Regulatory Sequence Prediction

[248] Predicting the function of regulatory elements from primary DNA sequence still remains a major problem in computational biology. These elements typically contain combinations of several binding sites for regulatory factors whose activity together specifies the developmental times, cell-types, or environmental signals in which the element will be active. Genetic variation in regulatory elements is increasingly thought to play a significant role in the etiology and heritability of common diseases, and surveys of Genome Wide Association Studies have highlighted the preponderance of significant variants in regulatory DNA. An accurate computational model to predict regulatory elements can 1) help identify and link core sets of regulatory factors with specific diseases, and 2) predict the functional consequences of variation or mutations in specific sites within regulatory elements.

[249] A method is disclosed herein for regulatory DNA sequence prediction, kmer-SVM, which uses combinations of short (6-8 bp) k-mer frequencies to predict the activity of larger functional genomic sequence elements, typically ranging from 500 to 2000bp in length. An advantage of k-mer based approaches relative to the alternative position weight matrix (PWM) approach is that PWMs can require large amounts of data to optimize and determine appropriate scoring thresholds, while k-mers are simple features which are either present or absent. However, in this previous implementation of the kmer-SVM, the choice to use a single k , and which k , is somewhat arbitrary and based on performance on a limited selection of datasets. This examples expands the single k approach to include longer and much more general sequence features. The function of these DNA regulatory elements is generally thought to be specified at the molecular level by the binding of combinations of Transcription Factors (TFs) or other DNA binding regulatory factors, and many of these binding sites are short and fall within the range of k (6-8) where the kmer-SVM approach was successful. However, Transcription Factor Binding Sites (TFBS) can vary from 6-20bp, so some are much longer (such as ABF1, CTCF, etc.), and thus cannot be completely represented by the short k-mers. Alternatively, TFBS can be defined by a set of sequences with some gaps (non-informative positions) as each given DNA sequence has some binding affinity for the TF. Although the kmer-SVM method can model TFBS longer than k by tiling across TFBS with overlapping k-mers, this loses some spatial information in the binding site, and overall

classification accuracy can be significantly impaired when long TFBS are important predictive features.

[250] One could address this issue by using longer k's or combinations of k-mers spanning the expected size range of TFBS, but a major limitation of this approach is that longer k-mers generate extremely sparse feature vectors (i.e. most k-mers simply do not appear in a training sequence and thus receive zero counts, or appear only once), which causes a severe overfitting problem even at quite moderate k. Therefore, the original kmer-SVM was limited in practice to k-mer lengths from 6 to 10, with performance already degrading at k = 9 or 10, depending on the dataset. Thus in practice, the parameter k was chosen by a tradeoff between resolving longer features and robust estimation of their frequencies.

[251] Gapped k-mers were a way to resolve this fundamental limitation with k-mer features and showed that they can be used to more robustly estimate k-mer frequencies in real biological sequences. Herein is disclosed a simple and efficient method for calculation of the robust k-mer count estimates. The kmer-SVM method was expanded to use gapped k-mers or robust k-mer count estimates as feature sets and present efficient methods to compute these new kernels. The method, gkm-SVM, consistently and significantly outperformed a kmer-SVM using both CTCF and EP300 genomic bound regions over a wide range of varying feature lengths. While kmer-SVM suffers significantly from overfitting as k is increased, gkm-SVM performance was only very modestly affected by changes in the chosen feature length parameters. The two approaches were compared on the complete human ENCODE ChIP-seq data sets, and showed that gkm-SVM either significantly outperformed or was comparable to kmer-SVM in all cases. Of biological interest, on the ENCODE ChIP-seq data sets, gkm-SVM outperformed the best known single PWM by detecting necessary co-factors. gkm-SVM was compared to similar earlier SVM approaches, and showed that they perform comparably for optimal parameters in terms of accuracy, but that gkm-SVM was less sensitive to parameter choice and was computationally more efficient. To further demonstrate the more general utility of the k-mer count estimates, they were applied in a simple Naïve-Bayes classifier, and showed that using k-mer count estimates instead of k-mer counts consistently improved classification accuracy. Since the method is general, many other sequence classification problems will also benefit from using these features. For example, word based methods can also be used to detect functional motifs in protein sequences, where the length of the functional domain is unknown.

Results

Calculation of sequence similarity score using gapped k-mers

[252] To overcome the limitations associated with using k-mers as features described above, gkm-SVM was developed which uses as features a full set of k-mers with gaps. At the heart of most classification methods is a distance or similarity score, often called a kernel function in the SVM context, which calculates the similarity between any two elements in the chosen feature space. Therefore, in this section, the feature set is described and how to efficiently calculate the similarity score. This new feature set, called gapped k-mers, was characterized by two parameters; (1) l , the whole word length including gaps, and (2) k , the number of informative, or non-gapped, positions in each word. The number of gaps is thus $l - k$.

First defined was a feature vector for a given sequence S to be $f^S = [y_1^S, y_2^S, \dots, y_M^S]^T$, where M is the number of all gapped k -mers (i.e. for DNA sequences, $M = \binom{l}{k} 4^k$), and y_i^S 's are

the counts of the corresponding gapped k -mers appeared in the sequence S . We then define a similarity score, or a kernel function, between two sequences, S_1 and S_2 , as the normalized inner product of the corresponding feature vectors as follows:

$$[253] \quad K(S_1, S_2) = \frac{\langle f^{S_1}, f^{S_2} \rangle}{\|f^{S_1}\| \|f^{S_2}\|} \quad (1)$$

[254] where $\langle f^{S_1}, f^{S_2} \rangle = \sum_{i=1}^M (y_i^{S_1} \cdot y_i^{S_2})$, and $\|f^S\| = \sqrt{\langle f^S, f^S \rangle}$. Therefore, the similarity score, $K(S_1, S_2)$, was always between 0 and 1, and $K(S, S)$ was equal to 1. Equation (1) is referred to as the gkm-kernel. It is similar to the wildcard kernel introduced in Leslie C, Kuang R (2004) Fast String Kernels using Inexact Matching for Protein Sequences. J Mach Learn Res 5: 1435–1455, but differs in that this method does not sum over the number of wild-cards, or gaps, as formulated in Leslie.

[255] Since the number of all possible gapped k -mers grows extremely rapidly as k increases, direct calculation of Equation (1) quickly becomes intractable. To implement gapped k -mers as features, it was necessary to overcome this serious issue, by deriving a new equation for $K(S_1, S_2)$ that does not involve the computation of all possible gapped k -mer counts. The key idea was that only the full l -mers present in the two sequences can contribute to the similarity score via all gapped k -mers derived from them. Thus the inner

product in Equation (1), which involves a sum over all gapped k -mers, can be computed by a much more compact sum, which involves only a double sum over the sequential l -mers present in each of the two sequences:

$$[256] \quad \langle f^{S_1}, f^{S_2} \rangle = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} h_{lk}(u_i^{S_1}, u_j^{S_2}) \quad (2)$$

[257] where $u_i^{S_1}$ was the i 'th l -mer appearing in S_1 , $u_j^{S_2}$ was the j 'th l -mer appearing in S_2 , and n_1 and n_2 were the numbers of full l -mers in S_1 and S_2 respectively, i.e. $n_1 = \text{length}(S_1) - l + 1$ and $n_2 = \text{length}(S_2) - l + 1$. Evaluation of Equation (2) was much more efficient than Equation (1) because almost always, $n_1 n_2 = M$. As will be shown below, $h_{lk}(u_1, u_2)$ only depends on the number of mismatches, m , between the two full l -mers, u_1 and u_2 , i.e. $h_{lk}(u_1, u_2) = h_{lk}(m)$. Therefore, Equation (2) was rewritten by grouping all the l -mer pairs of the same number of mismatches together as follows:

$$[258] \quad \langle f^{S_1}, f^{S_2} \rangle = \sum_{m=0}^l N_m(S_1, S_2) h_{lk}(m) \quad (3)$$

[259] where $N_m(S_1, S_2)$ was the number of pairs of l -mers with m mismatches, and $h_{lk}(m)$ was the corresponding coefficient. $N_m(S_1, S_2)$ was referred to as the *mismatch profile* of S_1 and S_2 . Since each l -mer pair with m mismatches contributes to $\binom{l-m}{k}$ common gapped k -mers, the coefficient $h_{lk}(m)$, denoted in short by h_m , is given by:

$$[260] \quad h_m = \begin{cases} \binom{l-m}{k} & l-m \geq k \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

[261] Determining a mismatch profile in Equation (3) was still computationally challenging since the numbers of mismatches between all possible l -mer pairs had yet to be determined. To address this issue, two different algorithms were developed. First, direct evaluation of the mismatch profiles between all pairs of training sequences was considered. To minimize the cost of counting mismatches between two words, an efficient mismatch counting algorithm was developed that practically runs in constant time, independent of k and l parameters (see Methods). Then Equation (3) was used to obtain the inner products for every pair of sequences.

[262] The direct and sequential evaluation of the kernel function between all training sequences became less practical as the number of training sequences got larger, since it required $O(N^2L^2)$ operations of mismatch counting between l -mer pairs, where N is the number of training sequences and L is the average sequence length. Because of this unfavorable scaling, an alternative method was implemented using a k -mer tree data structure, similar to one previously introduced in Leslie, above, but with some modifications (see Methods). This method simultaneously calculated the mismatch profile for all the sequence pairs, and, therefore, can significantly reduce the computation time especially when the number of gaps is relatively small, typically when $l - k \leq 4$. The efficiency was improved by truncating the sum in Equation (3) to only consider up to a maximum number of mismatches, m_{max} (see Methods). This approximate method was especially favorable when the number of gaps was large, but the efficiency comes at the cost of exact evaluation of the kernel and classification accuracy. One of the two algorithms was used, depending on the size of data sets and the number of gaps chosen for analysis.

Gapped k -mer SVM classifier outperforms k -mer SVM classifier

[263] Because of the difficulty of reliably estimating long k -mer counts, it was hypothesized that gkm-SVM would perform better than kmer-SVM, and that gapped k -mers would be most advantageous as features, when long TFBSs are important sequence elements in a given data set. To directly test this idea, the classification performance of gkm-SVM was compared to kmer-SVM in predicting the binding sites of CTCF (McDaniell R, Lee B-K, Song L, Liu Z, Boyle AP, et al. (2010) Heritable Individual-Specific and Allele-Specific Chromatin Signatures in Humans. *Science* 328: 235–239. doi:10.1126/science.1184655) in the human genome, a TF whose binding specificity has been well-characterized (Kim TH, Abdullaev ZK, Smith AD, Ching KA, Loukinov DI, et al. (2007) Analysis of the Vertebrate Insulator Protein CTCF-Binding Sites in the Human Genome. *Cell* 128: 1231–1245. doi:10.1016/j.cell.2006.12.048.). CTCF recognizes very long DNA sequences (the full PWM is 19bp), and the genomic CTCF bound regions are almost perfectly predicted by matches to the CTCF PWM in the PWM analysis, a predictor was used as the best matching log-odd score to the PWM model in the region, and achieved area under the ROC curve (AUC) of 0.983. It is very rare for a single PWM to perform this well, and CTCF may be unique in this regard. The CTCF dataset therefore provided an excellent opportunity to test the gapped k -mer classifier. The top 2,500 CTCF ChIP-seq signal enriched regions in the GM12878 cell

line available at Gene Expression Omnibus (GSE19622) (McDaniel, above) were used as a positive dataset, and equal numbers of random genomic sequences (1x) as a negative dataset. The negative sequences were generated by matching length, GC and repeat fraction of the positive set.

[264] The performance was compared of gkm-SVM and kmer-SVM on the CTCF data set for a range of oligomer lengths by varying either k (for kmer-SVM) or l (for gkm-SVM) from 6 to 20. The the parameter $k = 6$ for gkm-SVM was fixed. The classification performance of each was quantified by calculating test-set AUC with standard five-fold cross validation (CV) (see Methods). Figure 53A shows a summary of the comparisons. As anticipated, gkm-SVM performed consistently better than kmer-SVM for all lengths. More significantly, while kmer-SVM suffered severely from overfitting when k is greater than 10, gkm-SVM was virtually unaffected by l . In fact, gkm-SVM achieved the best result (AUC = 0.967) when $l = 14$ and $k = 6$, which was significantly better than the kmer-SVM (AUC = 0.912 when $k = 10$); the best ROC curve is shown in Figure 53C. It should be noted, however, that the PWM classification result is still the best (AUC = 0.983) among the three methods tested in this analysis. A complicating factor was that while both kmer-SVM and gkm-SVM used entire sequences (average length is 316 bp) to calculate the prediction scores, the PWM scores were from the best matching 19 bp sub-sequence in the region. It may be that the extra ~300 bp sequences contributed noise in the SVM prediction scores, which slightly impaired the overall classification accuracy. In any event, the gkm-SVM was a significant improvement in accuracy over the kmer-SVM, and both gkm-SVM and the PWM are excellent predictors on this dataset.

[265] Interestingly, gkm-SVM showed consistently better performance than kmer-SVM even if l is relatively small ($l < 10$) (Figure 53A). This suggested that gkm-SVM may also be better at modeling diverse combinations of TFBSs than kmer-SVM. To test this hypothesis, a mouse enhancer dataset of more varied sequence composition was analyzed: genomic EP300 bound regions in embryonic mouse forebrain (Visel A, Blow MJ, Zhang T, Akiyama JA, Holt A, et al. (2009) ChIP-seq accurately predicts tissue-specific activity of enhancers. Nature 457: 854–858. doi:10.1038/nature07730.) The original kmer-SVM classifiers can accurately predict EP300 binding when mediated by sets of active TFBSs (Lee D, Karchin R, Beer MA (2011) Discriminative prediction of mammalian enhancers from DNA sequence. Genome Res 21: 2167–2180. doi:10.1101/gr.121905.111.) This EP300 data set provided a direct test

of the effectiveness of using gapped k -mer features to detect more complex regulatory features. For this analysis, a new set was defined of the 1,693 400bp sites that maximize the EP300 ChIP-seq signal within each of the peaks determined by MACS (Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, et al. (2008) Model-based Analysis of ChIP-Seq (MACS). *Genome Biol* 9: R137. doi:10.1186/gb-2008-9-9-r137) after removing any regions which were more than 70% repeats. The k and l scaling were repeated with the EP300 data set and a 1x negative set, and again found that gkm-SVM consistently outperforms kmer-SVM for all feature lengths (Figure 53B). Analogous to the observations modeling CTCF binding, gkm-SVM AUC was high and did not degrade with large l . In contrast, the kmer-SVM accuracy dropped rapidly as k increased. Moreover, although the difference in performance was smaller than found for the CTCF data set, the gkm-SVM achieved the best AUC (0.947) with $l = 9$ and $k = 6$, while the kmer-SVM achieves 0.932 with $k = 7$, suggesting that longer k -mers with some flexibility do contain more complete information about TF binding (Figure 53D). At the same time, the gapped k -mer features were more robustly estimated (having more counts) and for this reason made more reliable predictors. The consequences of these improvements in AUC were significant when considering the genome-scale precision of the improved gkm-SVM classifiers. The rate of false positive predictions is dominated by the large neutral fraction of the genome, so the precision of a genome-scale classifier is best assessed by a Precision-Recall curve in combination with a much larger negative set, as discussed in Lee, 2011, above. For CTCF, at a recall of 50%, the precision increased from 36% to 59%. These ranges of precision and recall were in the relevant range of experiments aiming to discover and test novel enhancers, and it is expected that predictions based on gkm-SVM will have up to a two-fold higher successful validation rate.

[266] One further modification can substantially reduce the computational cost of using gapped k -mers with little degradation in performance. The algorithm using the k -mer tree data structure produced identical results to the direct evaluation of Equation (3), but typically was much faster when the number of mismatches, $l - k$, is smaller than four, and the number of training sequences is large. The k -mer tree algorithm can be made even more computationally efficient, the traversal of the tree is pruned by ignoring any k -mer pairs that have more mismatches than a predetermined parameter, m_{max} . This provided an approximation to the exact kernel calculation, but the approximation error was usually negligible given that the coefficient h_m for large numbers of mismatches were generally much smaller compared to those with small m . This approximation significantly reduced the total

number of calculations and allowed the user to control the running time of the algorithm by setting the parameter m_{max} , and makes the use of longer word lengths l feasible for any given k . To systematically investigate the classification performance of this approximation, the same analysis above was applied using both CTCF and EP300 data sets (Figure 53A, B), and found that AUCs from the approximate method were virtually identical to the exact method when the difference between m_{max} and $l - k$ are small. Interestingly, the approximation method achieved even higher AUC with CTCF data set in some cases.

[267] gkm-SVM to kmer-SVM were compared using a very broad range of human data sets generated by the ENCODE project (Gerstein MB, Kundaje A, Hariharan M, Landt SG, Yan K-K, et al. (2012) Architecture of the human regulatory network derived from ENCODE data. *Nature* 489: 91–100. doi:10.1038/nature11245; Wang J, Zhuang J, Iyer S, Lin X, Whitfield TW, et al. (2012) Sequence features and chromatin structure around the genomic regions bound by 119 human transcription factors. *Genome Res* 22: 1798–1812. doi:10.1101/gr.139105.112) 467 sets were used of ChIP-seq peaks produced by the ENCODE uniform processing pipeline containing at least 500 regions (see Methods). Any data set with greater than 5,000 regions by random sampling was truncated. Both kmer-SVM and gkm-SVM were trained on each set against an equal size (1x) negative set of random genomic regions and calculated AUCs with five-fold cross validation. We used $k = 6$ for kmer-SVM, and $l = 10$ and $k = 6$ for gkm-SVM, but as shown in Figure 53 the improvements are generally insensitive to these parameter choices. gkm-SVM almost always outperformed kmer-SVM (Figure 54A). Variances of AUCs from test CV sets were generally reduced, suggesting that gkm-SVM was more robust than kmer-SVM. gkm-SVM performed much better especially for TFs with long binding sites. In this dataset, most of these long binding sites arise in ChIP-seq data sets for CTCF and members of the cohesin complex (RAD21, SMC3) known to be physically associated with CTCF (Parelho V, Hadjur S, Spivakov M, Leleu M, Sauer S, et al. (2008) Cohesins Functionally Associate with CTCF on Mammalian Chromosome Arms. *Cell* 132: 422–433. doi:10.1016/j.cell.2008.01.011). On these CTCF associated factors, gkm-SVM exhibited much higher AUC than kmer-SVM, as highlighted by the cluster of circles (indicated by □) in Figure 54A. gkm-SVM was compared to the best single PWM AUC as shown in Fletez-Brant C, Lee D, McCallion AS, Beer MA (2013) kmer-SVM: a web server for identifying predictive regulatory sequence features in genomic data sets. *Nucleic Acids Res* 41: W544–W556. doi:10.1093/nar/gkt519, which is herein incorporated in its entirety, (Figure 54B). As expected, gkm-SVM

outperformed all datasets except CTCF, for which gkm-SVM performance was only marginally reduced. For a consistent analysis of this dataset, $l = 10$ and $k = 6$ was used, although for CTCF the gkm-SVM performance was optimal at larger l , as seen in Figure 53A.

Motif analysis of the ENCODE ChIP-seq data sets

[268] The predictive sequence features that allow gkm-SVM to outperform the single best PWM implied that cooperative binding is the underlying molecular mechanisms that targets TFs to these regulatory regions. Previously the focus was on a handful of the highest SVM weight k -mers (say top ten positive and top ten negative weight k -mers) to interpret the classification result. This simple method becomes unwieldy when applied to the gkm-SVM results because of the large number of very similar significant features (when l and/or k are large). Although the k -mers at the extreme top and bottom tails of the k -mer weight distribution are still important and biologically meaningful, those k -mers usually covered only a fraction of the significant feature set, and many more important features were included in the larger tails of the k -mer weight distribution. Therefore, more sophisticated algorithms were needed to extract the biologically relevant features from the classification results.

[269] A new method was developed to combine multiple similar k -mers into more compact and interpretable PWMs and analyzed the 467 ENCODE data sets (Wang, 2012 above). A larger number of predictive k -mers was used to build *de novo* PWMs (see Methods). The top 1% of 10-mers from each of the gkm-SVMs trained on the ENCODE data sets was used and identified up to three distinct PWMs from k -mers in this set. The results with the previous PWMs found in the same data sets were compared using a conventional tool (MEME-ChIP) Wang, 2012 and Machanick P, Bailey TL (2011) MEME-ChIP: motif analysis of large DNA datasets. *Bioinformatics* 27: 1696–1697. doi:10.1093/bioinformatics/btr189) Wang *et al.* analyzed 457 ENCODE ChIP-seq data sets (440 sets are in common with those analyzed above) and identified five PWMs from each data set. Collectively, Wang *et al.* found 79 distinct PWMs enriched, of which this method recovered 74. Comparing each ChIP-seq data set individually, this method recovered most of the PWMs reported by Wang *et al.* using the disclosed above method (Figure 54C). Interestingly, while Wang *et al.* largely failed to identify biologically meaningful PWMs from most of the POL2 ChIP-seq data sets (47 out of 58 sets returned no meaningful PWMs), our methods frequently identified cell-specific TFs

as well as promoter specific TFs. For example, the GATA1 TF identified from POL2 ChIP-seq in the erythroleukemic cell line K562 is known to play central roles in erythroid differentiation. The ETS1 TF from HUVEC is another extensively studied TF, known to be important for angiogenesis. A major difference between the two methods is the number of training sequences. The disclosed method used 10x larger numbers of ChIP-seq peaks (5,000 regions), and the large training sizes enabled identification of diverse combinatorial sequence features.

Comparison to previous kernels

[270] Since the early development of *k*-mer based supervised machine learning techniques (Leslie C, Eskin E, Noble WS (2002) The spectrum kernel: a string kernel for SVM protein classification. Pac Symp Biocomput: 564–575.), there have been a number of improvements. Some of these extend the feature set to include imperfect matches, similar in spirit to the gkm-SVM. The mismatch string kernel (Leslie C, Eskin E, Cohen A, Weston J, Noble WS (2004) Mismatch string kernels for discriminative protein classification. Bioinformatics 20: 467–476. doi:10.1093/bioinformatics/btg431.) is one such method, originally motivated by the fact that homologous protein sequences are not usually identical and have many frequently mutated positions. The mismatch kernel also uses *k*-mers as features, but allows some mismatches when counting *k*-mers and building feature vectors. The wildcard kernel (Leslie C, Kuang R (2004) Fast String Kernels using Inexact Matching for Protein Sequences. J Mach Learn Res 5: 1435–1455) is another variant of the original string kernel, which introduces a wildcard character that matches any single letter in the given alphabet. More recently, an alternative di-mismatch kernel (Agius P, Arvey A, Chang W, Noble WS, Leslie C (2010) High Resolution Models of Transcription Factor-DNA Affinities Improve In Vitro and In Vivo Binding Predictions. PLoS Comput Biol 6: e1000916. doi:10.1371/journal.pcbi.1000916.) has been proposed to directly model TFBSs, and has been successfully applied to protein binding microarray (PBM) data sets (Berger MF, Philippakis AA, Qureshi AM, He FS, Estep PW, et al. (2006) Compact, universal DNA microarrays to comprehensively determine transcription-factor binding site specificities. Nat Biotechnol 24: 1429–1435. doi:10.1038/nbt1246) and several other ChIP-seq data sets Agius P, Arvey A, Chang W, Noble WS, Leslie C (2010) High Resolution Models of Transcription Factor-DNA Affinities Improve In Vitro and In Vivo Binding Predictions. PLoS Comput Biol 6: e1000916. doi:10.1371/journal.pcbi.1000916; Arvey A, Agius P, Noble WS, Leslie C (2012) Sequence and chromatin determinants of cell-type-specific transcription factor

binding. *Genome Res* 22: 1723–1734. doi:10.1101/gr.127712.111) The di-mismatch method tries to overcome the limitation of the mismatch kernel by favoring k -mers with consecutive mismatches. However, in a recent comparison of methods for modeling transcription factor sequence specificity, full k -mer methods outperformed the di-nucleotide approaches when applied to PBM data (Weirauch MT, Cote A, Norel R, Annala M, Zhao Y, et al. (2013) Evaluation of methods for modeling transcription factor sequence specificity. *Nat Biotechnol* 31: 126–134. doi:10.1038/nbt.2486).

[271] The gkm-kernel was compared with the aforementioned three alternative methods, Mismatch kernel, Wildcard kernel, and Di-mismatch kernel, using the mouse forebrain EP300 data set. As shown in Figure 55, gkm-kernel outperformed the other three existing methods both in terms of the classification accuracy and running time. The best AUC achieved for gkm-kernel is 0.947 as compared to 0.937, 0.935, and 0.944 for the wildcard kernel, mismatch kernel, and di-mismatch kernel, respectively (Figure 55A). Although the wildcard kernel and gkm-kernel are quite similar, the systematic improvement in gkm-kernel AUCs was primarily due to the incorporation of reverse complement sequences. This was directly tested by adding reverse complement sequences to the feature set for the previously published methods, and indeed found that with this modification, these methods were also able to achieve comparable AUCs.

[272] More significantly, when running times were compared at parameters which maximize AUC for each method, the gkm-SVM implementation ($l=9, l-k=3$) was roughly two orders of magnitude faster than di-mismatch and slightly more than one order of magnitude faster than mismatch ($l=10, m=2$) and wildcard ($l=8, m=3$) on the EP300 data set (Figure 55B). Also, by fixing $k=6$ and the parameter m_{max} in the disclosed method's algorithm, the AUC becomes less sensitive to the feature length l , compared to a scan at fixed m , varying k (Figure 55A). Direct running time comparisons were made using the disclosed method's tree structure in the mismatch and wildcard kernels (described below). The di-mismatch kernel was tested up to $l = 10$, because it required more than 128GB of memory and did not finish within 2000 minutes when using $l = 11$.

[273] Both Mismatch kernel and Wildcard kernel are special cases of the more general class of kernels, defined by Equation (3). This unification allows direct application of the disclosed

methods herien for mismatch profile computation and therefore gives more efficient methods for computation of these existing methods (see Methods).

Calculation and performance of estimated l -mer frequencies for gkm-SVM

[274] As an alternative to the gapped k -mer feature set an alternative kernel was developed by replacing the k -mer counts with robust l -mer count estimates in the original kmer-SVM framework. Efficient methods to compute this new kernel are disclosed in Methods below. In Ghandi M, Mohammad-Noori M, Beer MA (2013) Robust k-mer frequency estimation using gapped k-mers. J Math Biol: 1–32. doi:10.1007/s00285-013-0705-3, the mapping from l -mers to gapped k -mers was considered. Among all possible sets of l -mer frequencies that could produce the same gapped k -mer frequency distribution, a method was developed to estimate the “most likely” l -mer frequency set. Full details of this method are described in Ghandi M, Mohammad-Noori M, Beer MA (2013) Robust k-mer frequency estimation using gapped k-mers. J Math Biol: 1–32. doi:10.1007/s00285-013-0705-3, incorporated by reference in its entirety. In brief, we first define a gapped k -mer count vector $f_{lk} = [y_1, y_2, \dots, y_M]$ similar to the definition of the gapped k -mer feature vector for gkm-SVM as shown above was defined. Then, the count estimate, $\hat{x}_u$, for l -mer u is given by

$$[275] \quad \hat{x}_u = \sum_{i=1}^M w_i y_i \quad (5)$$

[276] The weight w_i in Equation **Error! Reference source not found.** was shown to only depend on the number of mismatches, m , between the gapped k -mer corresponding to y_i and u , and takes the following form:

$$w(m) = \frac{(-1)^m}{b^\ell \binom{l}{k-m}} \frac{l-k}{l-k+m} \sum_{t=0}^{k-m} \binom{l}{t} (b-1)^t \quad (5)$$

[277] where b is the alphabet size and is equal to four in case of DNA sequences (A,C,G and T). Since the above equation is applied to every l -mer, it would provide a non-zero frequency even for an l -mer that does not have any exact match appearing in any training set sequence.

[278] Direct calculation of Equation **Error! Reference source not found.**, however, required actual counting of all of the M gapped k -mers, which becomes computationally intractable for large l and k in a way similar to Equation (1). Besides, summing up a large set

of floating point numbers may result in poor numerical precision. To overcome these issues, a simple method was developed, referred to as the *gkm-filter*, to more efficiently calculate the robust *l*-mer count estimates, $\hat{x}_u$, without calculating the intermediate gapped *k*-mer counts (see Methods). In summary, in the calculation of the robust *l*-mer count estimates, a non-zero weight was given to *l*-mers with few numbers of mismatches. The *k*-mer frequency estimation method was not constrained to produce non-negative frequencies and may occasionally generate negative count estimates. To obtain strictly positive frequencies, a revised version of the *gkm-filter* method was used, which is referred to as the *truncated gkm-filter*. Finally, a method was developed to directly calculate the kernels using these feature sets (see Methods). The evaluation of the *gkm-kernel* (the inner product of the *l*-mer count estimates vectors) is still given by Equation (3), but with a new set of weights $c_{lk}(m)$ given by Equation (13), below, replacing $h_{lk}(m)$. Therefore, efficient algorithms for pairwise mismatch profiles that were developed for the *gkm-kernel* can be directly used for this new feature set without any modification. Because of this symmetry, this method is referred to as *gkm-kernel with (full or truncated) filter*. See Ghandi, Mahmoud, Dongwon Lee, Morteza Mohammad-Noori, and Michael Beer, (2014) Enhanced Regulatory Sequence Prediction using Gapped k-mer Features, PLOS Computational Biology, 10(7), e1003711. Doi:10.1371/journal.pcbi.1003711, which is herein incorporated by reference in its entirety.

[279] To systematically compare the classification performance of these new methods with the original gapped *k*-mers, the previous analysis with the ENCODE ChIP-seq data sets was repeated. Using the truncated *gkm-filter* yielded results highly comparable to the original *gkm-SVM* for most datasets with modestly but consistently better relative performance when AUC is greater than 0.9. Any improvement in the range of high AUC (> 0.9) typically strongly reduced the classifier's False Prediction Rate (Lee D, Beer MA (2014) Mammalian Enhancer Prediction. Genome Analysis: Current Procedures and Applications. Caister Academic Press.), therefore, the truncated filter method is recommended as the method of choice for most analyses. Compared to the original *gkm-SVM*, using the *gkm-SVM* with full filter yields lower AUCs although it is still significantly higher compared to the *kmer-SVM* method.

Application of the robust l -mer count estimates for Naïve-Bayes classifier

[280] Gapped k -mer based methods for improving sequence kernel methods are disclosed. By direct use of gapped k -mers as features or by using the robust l -mer count estimates, the long k -mers' sparse count problem was overcome for these methods. The general utility of the robust l -mer count estimates in sequence classification problems was demonstrated by applying it to a simple Naïve-Bayes (NB) classifier similar to the one previously introduced in Sandberg R, Winberg G, Bränden C-I, Kaske A, Ernberg I, et al. (2001) Capturing Whole-Genome Characteristics in Short Sequences Using a Naïve Bayesian Classifier. *Genome Res* 11: 1404–1409. doi:10.1101/gr.186401 and show that by using robust count estimates instead of conventional k -mer counts the performance of the Naïve-Bayes classifier for long k -mers was boosted.

[281] Here, the log-likelihood ratio of the estimated l -mer frequencies in the positive and negative sets was used as a predictor, using the NB assumption of feature independence. The prediction score of any given sequence of length n , denoted by $S = s_0s_1\dots s_{n-1}$, is then given by:

$$f_{NB}(S) = \sum_{i=0}^{n-l} \log \frac{N_P(s_i s_{i+1} \dots s_{i+l-1})}{N_N(s_i s_{i+1} \dots s_{i+l-1})} \quad (6)$$

where N_P and N_N are the robust count estimates of the corresponding l -mers, $s_i s_{i+1} \dots s_{i+l-1}$, in the positive and negative training set, and are given by Equation (10) below. The *truncated* gkm-filter method was used adding pseudo-count (half of the smallest positive coefficient of the truncated gkm-filter) to each of the estimated frequencies to obtain strictly positive frequencies for log-likelihood ratio. As a comparison, the NB classifier was implemented without the gkm-filter, using actual l -mer counts with a pseudo-count (0.5) for N_P and N_N . It was predicted that the CTCF and EP300 genomic bound regions with both NB classifiers (i.e. with and without using robust count estimates). As shown earlier, genomic CTCF bound regions are almost perfectly predicted by the single CTCF PWM, and the local sequence features around the CTCF binding motif do not seem to significantly contribute to the prediction. Thus, to precisely detect the CTCF binding motif and achieve the best classification performance, every substring of length $n = 15 + l - 1$ was scored for each sequence and assigned the maximum as the final score for the sequence. The window size of 15 was chosen to optimize the detection of the CTCF site within a small window of flanking

sequence, which maximized the performance of the NB classifier without the gkm-filter. For the EP300 genomic bound regions, in contrast, the full sequence was used in both classifiers. The performance of these NB classifiers was compared on both data sets in Figure 56 for a range of feature length (6-20bp). Similar to the previous analysis using gkm-SVM and kmer-SVM (Figure 53), using robust count estimates (gkm-filter) significantly improved the classification accuracy especially for longer k -mers (Figure 56). On the CTCF data set, the NB classifier using the gkm-filter achieved best performance with $l = 20$ (AUC = 0.99), which is even better than that of the CTCF PWM (smallest dotted line, AUC = 0.983) (Figure 56A). Also on the EP300 dataset, the gkm-filter significantly improved the overall performance of NB classifier (Figure 56B). The superior classification performance using gapped k -mer based features was consistent for both SVM and NB classifiers, and strongly suggested that the robust l -mer count estimates provide a more complete and robust set of sequence features than simple k -mers in most sequence classification problems.

Discussion

[282] Disclosed in this example is a significantly improved method for sequence prediction using gapped k -mers as features, gkm-SVM. A new set of algorithms was disclosed to efficiently calculate the kernel matrix, and demonstrated that by using these new methods the sparse k -mer count problem for long k -mers was overcome and hence significantly improved the classification accuracy especially for long TFBSs. Detailed comparisons of the disclosed method with some existing methods showed that the gkm-SVM outperformed existing methods in terms of classification accuracy on benchmark data and was also typically orders of magnitude faster. The concept of gkm-filters for efficient calculation of the robust k -mer count estimates and derived optimal weights for penalizing different number of mismatches was introduced. It was shown that k -mers can be replaced with robust k -mer count estimates to avoid long k -mer sparse count problem, and demonstrated the effectiveness of this method by showing examples in SVM and Naïve-Bayes classifiers. Most k -mer based methods can be significantly improved by simply using this generalized k -mer count.

[283] The main biological relevance of the computational method disclosed in this Example is that gkm-SVM was capable of accurately predicting a wide range of specific classes of functional regulatory elements based on DNA sequence features in those elements alone. This implied that the epigenomic state of a DNA regulatory element primarily is specified by

its sequence. In addition, the predictions facilitate direct investigation of how these elements function, either by targeted mutation of the predictive elements within the larger regulatory region, or by modulating the activity of the TFs which bind the predictive sequence elements. Other Examples herein use changes in the gkm-SVM score to systematically evaluate the predicted impact of human regulatory variation (single nucleotide polymorphisms (SNPs) or indels) to interpret significant SNPs identified in genome wide association studies. The gkm-SVM was demonstrated to be better at predicting all ENCODE ChIP-seq data than the best single PWM found from the ChIP-seq regions, or previously known PWMs. The gkm-SVM was able to do so by integrating cofactor sequences which may not be directly bound by the ChIP-ed TF but facilitate its occupancy. To predict this ChIP-seq set accurately required the improved accuracy of the gkm-SVM and its ability to describe longer binding sites such as CTCF, which were very difficult for the earlier kmer-SVM approach. Most of the cofactors found by traditional PWM discovery methods were recovered, but it was shown that these combinations of cofactors are predictive in the sense that they are sufficient to define the experimentally bound regions.

[284] There are some further issues that need to be considered in the application of these methods. First, one will typically be interested in finding an optimal set of the parameters (l and k) to achieve the best classification performance. A significant advantage of gapped k -mer methods over k -mer methods is that they are more robust and are less sensitive to the particular choices of l or k compared to kmer-SVM or NB classifiers, as shown in Figure 53 and Figure 56. Nevertheless, these parameters can still be optimized to maximize cross validation AUC. As a general rule, when choosing the parameter k , which determines how different numbers of mismatches are weighted, given a whole word length l , smaller values of k (typically less than 8) are usually better when important sequence elements are believed to be more degenerate or when only small amount of training data is available. Although the choice of k directly affects the feature set, the analysis of several datasets showed that the overall performance of the classifier was not very sensitive to changes in k . The parameter l was directly related to TFBS lengths and should be comparable to or slightly larger than the longest important feature, as demonstrated by our analysis of the CTCF and EP300 data sets in Figure 53 and Figure 56.

[285] The disclosed method also avoids an issue that would arise if one chose instead to directly use Equation **Error! Reference source not found.** for computing count estimates.

This would involve a large number of floating point operations, and accumulated round-off error could become significant in the large summations. There are some algorithms, such as Kahan compensated summation, which can significantly reduce this error, however, evaluating this sum was avoided by first computing the mismatch profiles between sequences, which involves only integer calculations. Then, the weighted sum of the number of mismatches was calculated using Equation (10), which involves a much smaller number of floating point operations.

[286] Two issues which are left for future investigation are different treatment of end vs. internal gaps, and allowing imperfect mismatches. Special consideration for gaps which occur at the end of a k -mer instead of internal gaps were not made. Also, implementation of a mismatch treats all nucleotides equally, but often TF binding sites can prefer an A or T in a given position, or a purine vs. pyrimidine pair. The disclosed method recovers these preferences by assigning different weights to k -mers which do not have gaps at these positions, but including a wider alphabet including (W,S,Y,R) for (AT,GC,AG,CT) may have some advantages.

[287] This example focused on using DNA sequences as features for classifying the molecular or biological function of a genomic region. The method can be applied to any classification or prediction problem involving a large feature set. In general, when the number of features used by a classifier increases, the number of samples in the training set for each point in the feature space becomes smaller, and small sample count issues occur (which was resolved using gapped k -mers). One approach to the large feature space is *feature selection*, which selects a subset of features and builds a classifier only using those features, ignoring all the other features. However, usually a limited subset of features cannot explain all the variation in the predicted quantity. While hypothetical at this point, the disclosed analysis suggested that an alternative approach might be of general value. Analogous to the way gapped k -mers were used to more robustly estimate k -mer feature frequencies, there may be a general approach which uses subsets of a larger feature set to combine observed feature counts with weights reflecting the similarity to some generalized feature. These estimated feature frequencies will be less susceptible to statistical noise by construction, and thus may provide consistently better classification performance, as shown for gapped k -mers.

Methods

Support Vector Machine

[288] The Support Vector Machine (SVM) is one of the most successful binary classifiers and has been widely used in many classification problems. An SVM based framework, or “kmer-SVM” was previously developed, for enhancer prediction and have successfully applied to embryonic mouse enhancers and many other regulatory datasets. Briefly, the kmer-SVM method finds a decision boundary that maximally discriminates a set of regulatory sequences from random genomic non-regulatory sequences in the k -mer frequency feature vector space. In this Example, new kernel functions using gapped k -mers and l -mer count estimates as features were disclosed, and software that calculates the kernel matrix. For SVM training, a custom Python script was developed that takes the kernel matrix as input and learns support vectors. Shogun Machine Learning Toolbox (Sonnenburg S, Rätsch G, Henschel S, Widmer C, Behr J, et al. (2010) The SHOGUN Machine Learning Toolbox. J Mach Learn Res 11: 1799–1802.) and SVM-light (Joachims T (1999) Making large-scale support vector machine learning practical. Advances in Kernel Methods. Cambridge, MA: MIT Press. pp. 169–184.) was used for the SVM training script. As an alternative method, an SVM classifier based on the iterative algorithm described in (Jaakkola T, Diekhans M, Haussler D (2000) A Discriminative Framework for Detecting Remote Protein Homologies. J Comput Biol 7: 95–114. doi:10.1089/10665270050081405.) was implemented.

Direct Computation of Gkm-kernel

[289] For direct computation of the gkm-SVM kernel matrix, each training sequence was represented with a list of l -mers and corresponding count for each l -mer. Then for each pair of sequences, the number of mismatches was computed for all pairs of l -mers and used the corresponding coefficient h_m to obtain the inner product of Equation (3). As the number of unique l -mers in each sequence is L and the number of sequences is N , this algorithm would require $O(N^2L^2)$ comparisons. In addition, a naive algorithm for counting the number of mismatches between two l -mers (i.e. the hamming distance) would be $O(l)$. The implementation employed bitwise operators, providing a constant-factor speedup. Briefly, using two bits to represent each base (A,C,G and T), used was an integer variable to represent non-overlapping substrings of t base pairs of the l -mer, therefore using total $\left\lceil \frac{l}{t} \right\rceil$ integers to

represent each l -mer, where $\lceil \cdot \rceil$ is the ceiling function. For counting the number of mismatches, we take the bitwise XOR (exclusive OR) of the integer representations of the two l -mers and use a precomputed look-up table to obtain the total number of mismatches using the XOR result. This method required a look-up table of size 2^{2t} . The optimal value of t depends on the processor architecture and amount of cache memory. $t = 6$ was used for the analysis.

Gkm-kernel with k -mer Tree Data Structure

[290] As depicted in Figure 57, a k -mer tree was used to hold all the l -mers in the collection of all of the sequences. The tree was constructed by adding a path for every l -mer observed in a training sequence. Each node t_i at depth d represents a sub-sequence of length d , denoted by $s(t_i)$, which is determined by the path from the root of the tree to the node t_i . Each terminal leaf node of the tree represents an l -mer, and holds the list of training sequence labels in which that l -mer appeared and the number of times that l -mer appeared in each sequence. As an example, Figure 57 shows the tree that stores all the substrings of length $l=3$ in three sequences $S_1=AAACCC$, $S_2=ACC$, and $S_3=AAAAA$. Then, to evaluate the mismatch profile the tree was traversed in a depth-first search (DFS) order. In contrast to the mismatch tree used in (Leslie C, Eskin E, Cohen A, Weston J, Noble WS (2004) Mismatch string kernels for discriminative protein classification. *Bioinformatics* 20: 467–476. doi:10.1093/bioinformatics/btg431), here for each node t_i , at depth d , the list of pointers to all the nodes t_j at depth d for which $s(t_i)$ and $s(t_j)$ have at most $l - k$ number of mismatches was stored. Also stored were the number of mismatches between $s(t_i)$ and $s(t_j)$. Similar to the mismatch tree, there was no need to store these values for all the nodes in the tree, but they were computed recursively as the tree was traversed. When reaching a leaf node, the corresponding mismatch profile $N_m(S_i, S_j)$ was incremented for each pair of sequences S_i in that leaf node's sequence list, and all the S_j 's in the list of sequences in the pointer list for that leaf node. At the end of one DFS traversal of the tree, the mismatch profiles for all pairs of sequences were completely determined.

[291] To increase the speed further, an optional parameter m_{max} was introduced which limits the maximum number of mismatches. By setting m_{max} smaller than $l - k$, only considered were l -mer pairs that have at most m_{max} number of mismatches. This can reduce calculation significantly by ignoring l -mer pairs which potentially contribute less to the overall similarity

scores. This method provides fast and efficient approximations of the exact solution. In addition, only computed were the lower triangle of the matrix because of the symmetry in the kernel matrix. Hence, at each node t_i , excluded were the nodes t_j in the list that have $\text{maxID}(t_i) < \text{minID}(t_j)$, where $\text{minID}(t_i)$ and $\text{maxID}(t_j)$ are the maximum and minimum sequence ID in the subtrees of t_i and t_j respectively and are computed and stored for each node at the time the tree was built.

Analysis of *de novo* PWMs from gkm-SVM

[292] Disclosed is a method for building *de novo* PWMs by systematically merging the most predictive k -mers from a trained gkm-SVM. First determined was a set of predictive k -mers by scoring all possible 10-mers and selecting the top 1% of the high-scoring 10-mers. A set of distinct PWM models was found from these predictive 10-mers using a heuristic iterated greedy algorithm. Specifically, first built was an initial PWM model from the highest scoring 10-mer. Then, for each of the remaining predictive 10-mers, the log-odd ratios of all possible alignments of the 10-mer to the PWM model was calculated, and identified the best alignment (i.e. the position and the orientation that give rise to the highest log-odd ratio value). Since multiple distinct classes of TFBSs are expected to be identified in most cases, only considered were 10-mers with good alignments (i.e. a threshold of 5.0 for log-odd ratio scores relative to a genomic GC=0.42 background was used). After each of the 10-mers was aligned, the PWM model was updated only with successfully aligned 10-mers. To further refine the PWM, this was repeated by iterating through all of the top 1% 10-mers until no changes were made. When updating the PWM model, the assumption was that the contribution of each k -mer was exponentially weighted proportional to its SVM score, using $\exp(\alpha w_i)$, with $\alpha=3.0$. The 10-mers used for creating the 1st PWM were then removed from the list, and the process was repeated on the remaining predictive k -mers, to find up to three PWMs. Lastly, the PWMs were matched to the previously identified PWMs (Wang J, Zhuang J, Iyer S, Lin X, Whitfield TW, et al. (2012) Sequence features and chromatin structure around the genomic regions bound by 119 human transcription factors. *Genome Res* 22: 1798–1812. doi:10.1101/gr.139105.112.) using TOMTOM (Gupta S, Stamatoyannopoulos JA, Bailey TL, Noble WS (2007) Quantifying similarity between motifs. *Genome Biol* 8: R24. doi:10.1186/gb-2007-8-2-r24.) software. Each of the PWMs identified by the method were associated with Gupta PWMs if the q -value (false discovery rate) < 0.05 .

Implementation of Mismatch and Wildcard Kernels Using gkm-Kernel Framework

[293] In the gkm-kernel, the feature vector was defined to consist of the frequency of all the l -mers with exactly k known bases and $l - k$ gaps. In contrast, the wildcard kernel (above) also includes all the l -mers with $l - k$ wildcards, where $l - k$ ranges from 0 to the maximum number of wildcards allowed, M . Thus in the wildcard kernel, the parameter M replaces k in the disclosed gkm-kernel. In the sum, these are weighted by λ^{l-k} to penalize sequences with more wildcards. An equation was derived to directly compute the inner products from the mismatch profiles without the need to calculate the actual gapped k -mer counts. It is shown that a similar approach can be used to calculate the wildcard kernel. A new set of coefficients $h_{lM}^{wc}(m)$ was derived that can substitute h_m , in Equation (3). To evaluate $h_{lM}^{wc}(m)$ one only needs to consider the contribution of each pair of l -mers with m mismatches in the inner product of the corresponding feature vectors of the two sequences. Equation (7) gives those weights:

$$h_{lM}^{wc}(m) = \sum_{k'=l-M}^l \lambda^{l-k'} h_{lk'}(m) = \sum_{k'=l-M}^l \lambda^{l-k'} \binom{l-m}{k'} \quad (7)$$

[294] Using the above form allows direct use of the fast algorithms developed for calculation of the mismatch profiles to calculate the wildcard kernels. Although there are similarities between the disclosed tree algorithm and the tree algorithm described in mismatch above, there are some key differences. In the mismatch method, the algorithm literally transverses all the possible gapped l -mers (with maximum M number of gaps) while the disclosed algorithm takes advantage of the fact that the final inner product will only depend on the number of pairwise mismatches and hence only traverses all the l -mers that are present in the input data. Another difference is that the mismatch method uses a list of all partially matching l -mers at each node of the tree, while the disclosed method used a list of pointers to tree nodes instead. So, for example, at the beginning of the algorithm (at depth $d = 0$) the mismatch method started with a large list consisting of all the possible l -mers in the input data, while in the disclosed algorithm the list at depth $d = 0$ consisted of only one node (the root of the tree). Using this representation of all the partially matching l -mers, the disclosed method more efficiently performed the comparisons at each step of the algorithm when the tree is dense.

[295] In the mismatch string kernel described above, the feature vectors consist of the counts for all the l -mers with maximum distance M from the l -mers in the sequence. The disclosed approach above can be used to implement the mismatch kernel. Again, the only difference is in the set of weights used in Equation (3). To calculate the mismatch string kernel value for two sequences $h_{lk}(m)$ in Equation (3) was replaced by $h_{l,M}^{mismatch}(m)$:

$$h_{l,M}^{mismatch}(m) = \sum_{m_1=0}^M \sum_{m_2=0}^M \sum_{t=0}^M \binom{l-m}{t} (b-1)^t \binom{m}{r} (b-2)^r \binom{m-r}{m_1-t-r} \quad (8)$$

where b is the alphabet size ($b = 4$ for DNA sequences) and $r = m_1 + m_2 - m - 2t$. Given two l -mers x_1 and x_2 where x_1 and x_2 differ in exactly m places, the term inside the summations counts the number of all possible l -mers that exactly differ x_1 in m_1 places and x_2 in m_2 places t of which fall in the common $l-m$ bases of x_1 and x_2 . So the result of the summation is the number of all l -mers that differ x_1 and x_2 in at most M places. This form for the mismatch string kernel has the advantage that one can use equation (3) to compute the kernels by only having the mismatch profiles that can be computed more efficiently.

Gkm-Filters for Computation of the Robust l -mer Count Estimates

[296] To compute the l -mer count estimates by using Equation **Error! Reference source not found.**, one should first calculate the gapped k -mer counts, y_i , and then use Equation **Error! Reference source not found.** to combine the y_i with a weight corresponding to the number of mismatches, given by Equation (5). The mapping from observed l -mer counts to gapped k -mer counts is performed by the matrix A , whose elements are a_{ij} . If the gapped k -mer v_i matches l -mer u_j , then $a_{ij} = 1$, otherwise $a_{ij} = 0$. There is a second matrix W , which performs the mapping from gapped k -mer counts to *estimated* l -mer counts, and whose elements are w_{ij} . It has been shown that matrix W is the Penrose-Moore pseudo-inverse of A . The element w_{ij} only depends on the number of mismatches between the l -mer u_i and the gapped k -mer v_j , and is given by Equation (5). For efficient computation, the two mapping matrices, A and W can be combined, and directly calculate the minimum norm l -mer count estimates from actual l -mer counts in a sequence. This combined mapping was referred to as the *gkm-filter*. The combined mapping matrix $G = WA$, has elements g_{ij} . As shown below, g_{ij} also only depended on the number of mismatches, m , between the l -mers u_i and u_j . These values are denoted by $g_{lk}(m)$ and refer to this as the *gkm-filter* since the domain and range of this mapping is the same.

[297] To obtain the element $g_{lk}(m)$, that gives the weight for the contribution of an observed l -mer u_i in the training set to the minimum norm l -mer count estimate u_j that has exactly m mismatches with u_i , one can sum over the contribution of all the gapped k -mers v_t that match u_i . Note that $\alpha_{ij} = 0$ for all other gapped k -mers. There exist $\binom{l-m}{k-t} \binom{m}{t}$ different gapped k -mers that match u_i and have exactly m mismatches with u_j . For a gapped k -mer to have exactly t mismatches with u_j , there are $\binom{m}{t}$ ways to select the t mismatch positions and $\binom{l-m}{k-t}$ ways to select the $k-t$ match positions. Now considering the weight $w(t)$ for the gapped k -mers with t mismatches, the gapped k -mer filter elements, $g_{lk}(m)$ can be obtained as follows:

$$g_{lk}(m) = \sum_{t=0}^m \binom{l-m}{k-t} \binom{m}{t} w(t) \quad (9)$$

In other words, there are $\binom{l-m}{k-t} \binom{m}{t}$ different ways to construct a gapped k -mer that matches u_i , and has exactly t mismatches with u_j , by selecting t positions from the m mismatch positions and $k-t$ positions from the $l-m$ match positions as explained above (Figure S11). It can be easily shown that $g_{lk}(m)$ is a polynomial of degree k in m . Now using the weights given in Equation (9), for any given l -mer, u finally obtained is the minimum norm l -mer count estimate as follows:

$$\hat{x} = \sum_{m=0}^l N_{tr}(u, m) g_{lk}(m) , \quad (10)$$

where $N_{tr}(u, m)$ is the number of l -mers with exactly m mismatches with u in the training set. For large values of l and k , the number of all possible gapped k -mers gets exponentially large and since this method avoids evaluating the gapped k -mer counts, it significantly reduces the cost of calculating the l -mer count estimates compared to the original method developed in Ghandi M, Mohammad-Noori M, Beer MA (2013) Robust k-mer frequency estimation using gapped k-mers. J Math Biol: 1–32. doi:10.1007/s00285-013-0705-3.

[298] In summary, a generalized k -mer count (referred to as the robust l -mer count estimates) was defined by giving a non-zero weight to l -mers with few number of mismatches (In the conventional k -mer count only perfectly matching k -mers are counted). These weights were given by $g_{lk}(m)$. Plots for $g_{lk}(m)$ for $l = 20$ and various values of k were made. Each plot was normalized so that weight corresponding to zero mismatches was equal to one. The case with $l = k$ is equivalent to the conventional k -mer count. Plots were made for $g_{lk}(m)$ for $k = 6$ and various values of l . With a fixed length l , higher values of k resulted in smaller coefficients for larger mismatches, and therefore less smoothing of the estimated counts. Moreover, $g_{lk}(m)$ can become slightly negative for large numbers of mismatches. This is because in the estimation of the frequencies the frequencies to be positive were not restricted, and doing so would yield a more complicated expression. The assumed Gaussian distribution allowed non-physical negative frequencies to have non-zero probability. A beta-distribution would not have this problem but would introduce offsetting complications. In cases where the estimated counts are required to be strictly positive, such as when there is a need to calculate the logarithm or ratios of the estimated frequencies, the gkm-filter $g_{lk}(m)$ can be truncated by setting $g_{lk}(m) = 0$ for every $m \geq m_0$, where m_0 is the smallest number of mismatches for which $g_{lk}(m_0) < 0$. This will give an approximation to the value of $\hat{x}$ in Equation **Error! Reference source not found.**, so it will no longer strictly be the minimum norm estimate, but it will guarantee that all the count estimates are non-negative.

Gkm-kernel with l -mer count estimates

[299] Given a sequence S , an l -mer count estimate vector $f^S = (\hat{x}_1^S, \hat{x}_2^S, \dots, \hat{x}_N^S)^*$ is defined where N is the number of all l -mers (4^l in case of DNA sequences), and $\hat{x}_i^S$ is the estimated count of the i^{th} l -mer appearing in sequence S using Equation (10). Then, calculated is a standard linear kernel simply by using this vector in Equation (1). Similar to the gkm-kernel method, this equation can be simplified using the same technique introduced in Equation (2) which does not involve the computation of individual l -mer estimates. It is shown that the inner product of the two l -mer count estimate vectors can be obtained as follows:

$$\langle f^{S_1}, f^{S_2} \rangle = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} c(u_i^{S_1}, u_j^{S_2}) \quad (11)$$

where n_1 and n_2 are the number of l -mers in S_1 and S_2 , and $u_i^{S_1}$ is the i 'th l -mer in S_1 and $u_j^{S_2}$ is the j 'th l -mer in S_2 . If u_1 and u_2 have exactly m mismatches then $c(u_1, u_2) = c_m$. Grouping all the l -mer pairs with m mismatches, rewrites Equation (11) as follows:

$$\langle f^{S_1}, f^{S_2} \rangle = \sum_{m=0}^l N_m(S_1, S_2) c_{lk}(m) \quad (12)$$

where $N_m(S_1, S_2)$ is the mismatch profile of S_1 and S_2 as previously defined in Equation (3). It is shown that the weight $c_{lk}(m)$, denoted in short by c_m , can be obtained as:

$$c_m = \sum_{m_1} \sum_{m_2} \sum_t \binom{l-m}{t} \binom{m}{m_1-t} \binom{m_1-t}{r} (b-1)^t (b-2)^r g_{m_1} g_{m_2} \quad (13)$$

where $r = m_1 + m_2 - 2t - m$, b is the alphabet size. The summations are taken over the range 0 to l . Given two l -mers u_1 and u_2 , with m mismatches and $l - m$ matched positions, enumerated was the number of all possible l -mers, u , that have m_1 mismatches with u_1 and m_2 mismatches with u_2 . For this, it was assumed that t of the m_1 mismatches are among the $l - m$ match positions and $m_1 - t$ of them are among the m mismatch positions. There are

$\binom{l-m}{t} \binom{m}{m_1-t}$ ways to choose these m_1 positions and $(b-1)^t$ choices for the values of the t

mismatches. These t mismatches plus the $m - (m_1 - t)$ unselected mismatch positions also do not match u_2 . Then, for the remaining $r = m_2 - (t + m - (m_1 - t))$ mismatches for u_2 there are

$\binom{m_1-t}{r}$ ways to select the positions and $(b-2)^r$ ways to select the values. Hence the total

number of l -mers, u with m_1 mismatches with u_1 and m_2 mismatches with u_2 , where t of the mismatches of u_1 and u are among the $l - m$ match positions of u_1 and u_2 is given by

$$\binom{l-m}{t} \binom{m}{m_1-t} \binom{m_1-t}{r} (b-1)^t (b-2)^r.$$

Using matrix notation, it was shown that $c_m = g_m$ if the full filter $g_{lk}(m)$ was used. To see this, note that $\langle f^{S_1}, f^{S_2} \rangle = f^{S_1} \cdot f^{S_2} = (G\mathbf{x}_1) \cdot G\mathbf{x}_2 = \mathbf{x}_1 G \cdot G\mathbf{x}_2$ where $\mathbf{x}_1$ and $\mathbf{x}_2$ are the l -mer count vectors for S_1 and S_2 . Given $G = WA$, thus $G \cdot G = (WA) \cdot WA = WAWA = WA = G$. Hence, $\langle f^{S_1}, f^{S_2} \rangle = \mathbf{x}_1 G\mathbf{x}_2$. Here A is the binary incidence matrix that maps l -mer counts to gapped k -mer counts as defined in Ghandi M, Mohammad-Noori M, Beer MA (2013) Robust k -mer frequency estimation using gapped k -mers. J Math Biol: 1–32. doi:10.1007/s00285-013-

0705-3. and W is the Moore-Penrose pseudo-inverse of A . Note that this result does not hold for the truncated filter g_m . In that case, directly use Equation (13) to obtain c_m coefficients.

ROC Curves

[300] To compare the performance of different classification methods, the area under the receiver operating characteristic (ROC) curve was calculated for each classifier. To plot the ROC curves and calculate area under the curves (AUCs) the ROCR package Sing T, Sander O, Beerenwinkel N, Lengauer T (2005) ROCR: visualizing classifier performance in R. *Bioinformatics* 21: 3940–3941. doi:10.1093/bioinformatics/bti623 in R was used.

Cross Validation

[301] Following standard five-fold cross validation procedures, the positive and negative sets were divided into five segments, left one segment out as the test set and used the other four segments for training. This was repeated for all of the five segments and calculated the mean and standard error of the prediction accuracy on the test set elements.

ENCODE ChIP-seq Datasets

[302] The ENCODE ChIP-seq datasets were downloaded from ftp://ftp.ebi.ac.uk/pub/databases/ensembl/encode/integration_data_jan2011/byDataType/peaks/jan2011/spp/optimal/hub/.

Implementation and Source Code

[303] These algorithms were implemented in C++, and the source code and executable files are available on the website at beerlab.org/gkmsvm/.

EXAMPLE 9 Predicting the Impact of Regulatory Variants From DNA Sequence

[304] Most variants implicated in common human disease by Genome-Wide Association Studies (GWAS) lie in non-coding sequence intervals (Hindorff, L. A. *et al.* Potential etiologic and functional implications of genome-wide association loci for human diseases and traits. *Proc. Natl. Acad. Sci. U. S. A.* 106, 9362–9367 (2009)). Despite the suggestion that regulatory element disruption represents a common theme, identifying causal risk variants

within indicted genomic regions remains a significant challenge. Disclosed herein is a novel sequence-based computational method to predict the effect of regulatory variation, using a classifier (gkm-SVM) disclosed in Example 8, which encoded cell-specific regulatory sequence vocabularies. The induced change in the gkm-SVM score, deltaSVM, quantified the effect of variants. The deltaSVM accurately predicted the impact of SNPs on DNase I sensitivity in their native genomic context, and accurately predicted the results of dense mutagenesis of several enhancers in reporter assays. Previously validated GWAS SNPs yield large deltaSVM scores, and the method disclosed herein predicted novel risk SNPs for several autoimmune diseases (See Fig. 5251A-F). The method and system comprising deltaSVM provides a powerful computational approach for systematically identifying functional regulatory variants.

Materials and Methods:

gkm-SVM and deltaSVM

[305] A gkm-SVM was trained by following previously reported methods with minor modifications (Ghandi, M., Lee, D., Mohammad-Noori, M. & Beer, M. A. Enhanced Regulatory Sequence Prediction Using Gapped k-mer Features. *PLoS Comput Biol* 10, e1003711 (2014); Lee, D., Karchin, R. & Beer, M. A. Discriminative prediction of mammalian enhancers from DNA sequence. *Genome Res.* 21, 2167–2180 (2011); Fletez-Brant, C., Lee, D., McCallion, A. S. & Beer, M. A. kmer-SVM: a web server for identifying predictive regulatory sequence features in genomic data sets. *Nucleic Acids Res.* 41, W544–W556 (2013); Gorkin, D. U. *et al.* Integration of ChIP-seq and machine learning reveals enhancers and a predictive regulatory sequence vocabulary in melanocytes. *Genome Res.* 22, 2290–2301 (2012)). Briefly, positive training set was defined by using publically available DNaseI-seq and ChIP-seq datasets, as discussed in greater detail below. A negative training set was then generated by randomly sampling from the genome equal number of regions that match length, GC and repeat fractions of the positive set. To remove false negative regions as much as possible, excluded were any regions with $P < 1e-5$ (MACS⁴³) from sampling. A gkm-SVM was trained with default parameters ($l = 10$, $k = 6$, and $d = 3$ with truncated filter), and measured the classification performance using ROC curves with five-fold cross validation. Scaling of performance with gkm-SVM feature length was performed. To calculate deltaSVM, 10-mer SVM scores were used as a proxy for weights. The final

weights were generated by averaging gkm-SVMs trained on five independently generated 1x negative sets. All gkm-SVM weights and source code used in this study are available at <http://www.beerlab.org/deltasvm>. When the deltaSVM between different training sets was compared, weights were normalized by the standard deviation of the weight distribution, but raw weights were reported here for simplicity. This correction is typically a small effect (<50%).

Training set for DNaseI Hypersensitive regions in lymphoblastoid cell lines

[306] GM12878 DNaseI-seq peaks were first defined by MACS⁴³ ($P < 1e-9$) for each replicate independently. Then peaks were chosen that were consistently found in both replicates. These peaks were further trimmed and 300bp central DHSs that maximize the DNase I hypersensitive signals were determined. Excluded were any regions with repeats > 70% and regions overlapping with dsQTLs, to avoid possible overfitting when scoring dsQTLs. Ultimately obtained was 22,384 300bp DHSs as the positive training set.

Training set for mouse melanocyte enhancers

[307] To train gkm-SVM appropriate for Tyr and Tyrp1 enhancers in mouse melanocytes, 4,337 EP300 bound regions were determined in the mouse melanocyte cell line melan-Ink4a-Arf (Gorkin, D. U. *et al.* Integration of ChIP-seq and machine learning reveals enhancers and a predictive regulatory sequence vocabulary in melanocytes. *Genome Res.* **22**, 2290–2301 (2012).) as the positive training set by following the above protocol with some adjustments (MACS $P < 0.002$). Promoter proximal regions and repeats were excluded from the training set. Since this positive set was much smaller than the others, 10x larger negative sets were generated in order to obtain more robust weights for deltaSVM analysis.

Training set for mouse liver enhancers

[308] Similar to the training set for DHSs in LCLs, a positive training set ($n = 19,590$) relevant to the ALDOB enhancer was defined by integrating DNaseI-seq and H3K4me1 ChIP-seq on adult mouse liver tissue (Yue, F. *et al.* A comparative encyclopedia of DNA elements in the mouse genome. *Nature* 515, 355–364 (2014)). To specify liver enhancers, additionally excluded were all promoter proximal DHSs (defined as regions with distances to

the nearest known transcription start sites (TSS) < 2kbp) from the training set, after determining the 300bp core DHSs as described above. Further selected were DHSs that overlap with H3K4me1 ChIP-seq peaks, which are well-known markers for enhancer activity (Heintzman, N. D. *et al.* Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome. *Nat Genet* 39, 311–318 (2007); Heintzman, N. D. *et al.* Histone modifications at human enhancers reflect global cell-type-specific gene expression. *Nature* 459, 108–112 (2009)), and defined these as the positive training set.

DeltaSVM analysis of dsQTL SNPs

[309] The dsQTL tables and the raw data files downloaded from the GEO database (accession number GSE31388) were used to define the positive and control sets of dsQTL SNPs. Because association alone does not necessarily imply the causation due in part to LD problem, more stringent rules were applied to determine the *most likely* causal dsQTL SNPs. First, the method was restricted to 1,296 SNPs within their associated 100bp DHSs to ensure that the changes in DNase I sensitivities are physically linked to the changes in their DNA sequences. Also applied was a more strict association P-value threshold ($P < 1e-5$) to reduce false positive associations, finally resulting in 579 SNPs. As a control SNP set, a 50x larger set of common random SNPs ($N = 28,950$; minor allele frequency > 5%) sampled only from the top 5% DHS regions that had been used to identify dsQTLs in a previous study (Degner, J. F. *et al.* DNase I sensitivity QTLs are a major determinant of human expression variation. *Nature* 482, 390–394 (2012)) was generated. To reduce false negative SNPs, excluded from sampling was any DHSs that had been found to be significantly associated with any of the dsQTL SNPs. Weights from gkm-SVM and kmer-SVM trained on the GM12878 DHSs were then used to calculate deltaSVM scores. Training a gkm-SVM on negative sequences constrained to match the positive sequences distance to TSS distribution did not affect overall performance was confirmed. Further confirmed was that using negative dsQTL control SNPs constrained to match the positive dsQTL distance to TSS and LD distribution did not affect overall performance. As a comparison, we considered three different scoring metrics; Combined-Annotation-Dependent Depletion (CADD, Kircher, M. *et al.* A general framework for estimating the relative pathogenicity of human genetic variants. *Nat. Genet.* **46**, 310–315 (2014)), Genome-Wide Annotation of Variants (GWAVA, Ritchie, G. R. S., Dunham, I., Zeggini, E. & Flicek, P. Functional annotation of noncoding sequence variants. *Nat. Methods* **11**, 294–296 (2014)), and conservation scores (Genomic Evolutionary Rate Profiling: GERP,

Davydov, E. V. *et al.* Identifying a High Fraction of the Human Genome to be under Selective Constraint Using GERP++. *PLoS Comput Biol* **6**, e1001025 (2010)). Pre-computed CADD scores for all 1000 Genome variants were downloaded (<http://cadd.gs.washington.edu>), from which the scores for the dsQTLs and control SNPs were extracted. Also extracted were the corresponding GWAVA scores from the pre-calculated table downloaded from the website (<ftp://ftp.sanger.ac.uk/pub/resources/software/gwava/>). All three different GWAVA models (region, tss, and unmatched) were analyzed and the best one (region) was chosen, as determined by AUC, for the main analysis. The GERP scores were also extracted from the same GWAVA result files. To do a fair comparison, only SNPs for which all the five scores are available were considered, resulting in 574 positive SNPs and 27,735 control SNPs. The entire prediction results are available in Table 13. eQTL beta was calculated using quantile-normalized gene expression available at http://eqtl.uchicago.edu/RNA_Seq_data/results.

Melanocyte Luciferase Assay and deltaSVM analysis

[310] 22 and 23 SNVs were selected for functional testing in the Tyr (mm10 coordinates chr7: 87508164-87508388; 226 bp) and Tyrp1 (mm10 coordinates chr4:80819561-80819851; 291 bp), respectively. These SNVs were randomly selected as follows: 10 SNVs in each enhancer predicted to reduce the enhancer's activity (negative deltaSVM), 4 SNVs in each enhancer predicted to increase the enhancer's activity (positive deltaSVM), 4 in each enhancer SNVs predicted to have a neutral impact on the enhancer's activity (deltaSVM near 0), and 4 (Tyr) or 5 (Tyr) additional SNVs that overlap with key motifs identified in previous reports^{20,21} (Murisier, F., Guichard, S. & Beermann, F. A conserved transcriptional enhancer that specifies Tyrp1 expression to melanocytes. *Dev. Biol.* 298, 644–655 (2006); Murisier, F., Guichard, S. & Beermann, F. The tyrosinase enhancer is activated by Sox10 and Mitf in mouse melanocytes. *Pigment Cell Res. Spons. Eur. Soc. Pigment Cell Res. Int. Pigment Cell Soc.* 20, 173–184 (2007)) Reference and SNV enhancer sequences were synthesized (Genewiz; South Plainfield, NJ), verified by sanger sequencing, and cloned into a luciferase reporter plasmid containing a minimal promoter and a luciferase reporter gene. For each SNV, 4 biological replicates (each with an independent plasmid DNA clone) were performed in order to control for differences that might arise from random mutations in the plasmid backbone or from variation in the quality of plasmid preps. Each reporter plasmid was transfected into the mouse melanocyte cell line melan-Ink4a-Arf, and measured luciferase

activity 24 hours later using the Dual-Luciferase Reporter Assay System (Promega; Madison, WI). The activity of each variant enhancer sequence was compared to the activity of the reference sequence (normalized to 1), and were thus able to quantitate the impact of each SNV on the enhancer's activity.

deltaSVM Analysis of Massively Parallel Reporter Assays

[311] To compare with exhaustive single nucleotide mutagenesis of the ALDOB enhancer (Patwardhan, R. P. *et al.* Massively parallel functional dissection of mammalian enhancers in vivo. *Nat. Biotechnol.* 30, 265–270 (2012)), a gkm-SVM was trained on adult mouse liver DHS as described above and scored each single nucleotide variant with deltaSVM and compared them with its measured *in vivo* expression changes. To compare with the directed mutagenesis of putative K562 and HepG2 enhancers (Kheradpour, P. *et al.* Systematic dissection of regulatory motifs in 2000 predicted human enhancers using a massively parallel reporter assay. *Genome Res.* 23, 800–811 (2013)), K562 and HepG2 specific gkm-SVMs were trained on the top 10000 500bp DHS regions in K562 and HepG2 cells (ENCODE Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature* 489, 57–74 (2012).), after excluding regions that were DHS in more than 30% of human ENCODE cell lines, or near promoters (<2kb from TSS), against an equal size GC and repeat matched training set. deltaSVM and the expression change were compared for pair of mutant wild-type constructs for each wild-type construct significantly expressed in either cell line (mean normalized expression > 3.5) which yielded 175 wild-type constructs and 277 mutant constructs: 102 of these are single base pair mutations and 175 are motif scrambling (8-17bp changed). For the motif scrambling mutations all 10-mer scores spanning the mutated motif were summed.

Training set for validated enhancers

[312] For each appropriate cell line, the top 10000 500bp DHS regions were trained on, after excluding regions that were DHS in more than 30% of human/mouse ENCODE cell lines/tissues, or near promoters (<2kb from TSS), against an equal size GC and repeat matched training set. The cell lines chosen were human LNCaP (ENCODE Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature* 489, 57–74 (2012)) for *Rfx6*, mouse erythroleukemia (MEL) (Yue, F. *et al.* A comparative encyclopedia of DNA

elements in the mouse genome. *Nature* 515, 355–364 (2014)) cells for *Bcl11a*, and HepG2(ENCODE Consortium, above) cells for *Sort1*.

Scoring of Autoimmune variants

[313] 11 autoimmune traits enriched in Th1 H3K27Ac were selected as shown in Fig 3 of Farh, K. K.-H. *et al.* Genetic and epigenetic fine mapping of causal autoimmune disease variants. *Nature* 518, 337–343 (2015). Predictions were made for 413 lead SNPs associated with 11 autoimmune diseases enriched in Th1 H3K27Ac regions (T1D: Type 1 Diabetes, CRO: Crohn's Disease, MS: Multiple Sclerosis, CEL: Celiac Disease, PBC: Primary Biliary Cirrhosis, RA: Rheumatoid Arthritis, Allergy, ATD: Autoimmune Thyroid Disease, UC: Ulcerative Colitis, VIT: Vitiligo, SLE: Systemic Lupus Erythematosus) (Farh, K. K.-H. *et al.* Genetic and epigenetic fine mapping of causal autoimmune disease variants. *Nature* 518, 337–343 (2015) A gkm-SVM was trained on the top 10000 500bp Th1 DHS regions, after excluding regions that were DHS in more than 30% of human ENCODE cell lines, or near promoters (<2kb from TSS), against an equal size GC and repeat matched training set. The lead SNP and all flanking off-lead candidates in LD as defined by ($R^2 > .5$ and PICS²⁸ probability>.0275) were scored, yielding 3113 total SNPs. Since the significance of the maximum deltaSVM score in a locus will depend on the number of SNPs in that locus, as a random control random SNPs and equal size flanking sets were scored. To determine the cutoff, first determined was the 2nd percentile deltaSVM score from 10,000 random permutations for each number of flanking SNPs (1~30), and then calculated mean and standard deviation of the 100 repeated experiments as the final cutoff. 17 high scoring deltaSVM SNPs were identified which were predicted to be expression perturbing SNPs with high confidence ($P < .02$), while at this threshold random sampling produced 8 SNPs (binomial test $P < 0.004$). See Fig. 52.

CLAIMS

What is claimed is:

1. A computer-implemented method for identifying nucleic acid variant sequences, comprising,
 - a) training a support vector machine classifier to generate a set of ranked gapped kmer-SVM weights;
 - b) establishing a scoring function characterized by a set of weights quantifying the contribution each possible variant
 - c) calculate a deltaSVM by summing the change in weight between alleles of each of the gapped-k-mer encompassing the variant
 - d) identifying predictive variant sequences from the calculated deltaSVM.
2. The method of Claim 1, further comprising diagnosing a subject having the variant sequence as having sequences indicating a disease or pathology.
3. The method of Claim 2, further comprising treating the diagnosed subject for the disease or pathology.
4. The method of Claim 1, wherein training a support vector machine comprises
 - i) providing a positive sequence set; and
 - ii) providing a negative sequence set;
5. The method of Claim 4, wherein the positive sequence set is provided from sequences of a subject.
6. The method of Claim 5, wherein the negative sequence set is matched to the positive sequence set profile by GC content, length, and repeat fraction from known nucleic acid sequences.
7. The method of Claim 6, wherein the negative sequence set is generated by random sampling.

8. The method of Claim 1, further comprising testing the predictive variant sequence in *in vitro* assays, *in vivo* assays, or both.
9. The method of Claim 1, wherein the predictive sequences are SNP (single nucleotide polymorphism), insertions, or deletions, or regulatory sequences.
10. The method of Claim 9, wherein the regulatory sequences are enhancer sequences, repressor sequences or insulator sequences.
11. The method of Claim 1, further comprising using the predictive variant sequences to diagnose a disease or pathology in a subject.
12. The method of Claim 11, wherein the disease or pathology is an autoimmune disease.
13. The method of Claim 12, wherein the autoimmune disease is Type 1 Diabetes, Crohn's Disease, Multiple Sclerosis, Celiac Disease, Primary Biliary Cirrhosis, Rheumatoid Arthritis, Allergy, Autoimmune Thyroid Disease, Ulcerative Colitis, Vitiligo, or Systemic Lupus Erythematosus.
14. A computer program comprising machine-executable instructions to cause a computer system to implement a method for identifying nucleic acid variant sequences, comprising,
 - a) providing a positive sequence set;
 - b) providing a negative sequence set;
 - c) training a support vector machine classifier to generate a set of gapped kmer-SVM weights; and
 - d) establishing a scoring function characterized by a set of weights quantifying the contribution each possible variant
 - e) calculate a deltaSVM by summing the change in weight between alleles of each of the gapped-k-mer encompassing the variant
 - f) identifying predictive variant sequences from the calculated deltaSVM.
15. A computer system for implementing a method for nucleic acid regulatory sequences, comprising,

a processing unit operable to

- a) provide a positive sequence set having a sequence profile;
- b) provide a negative sequence set;
- c) train a support vector machine classifier to generate a set of gapped kmer-SVM weights; and
- d) establish a scoring function characterized by a set of weights quantifying the contribution each possible variant
- e) calculate a deltaSVM by summing the change in weight between alleles of each of the gapped-k-mer encompassing the variant
- f) identifying predictive variant sequences from the calculated deltaSVM.

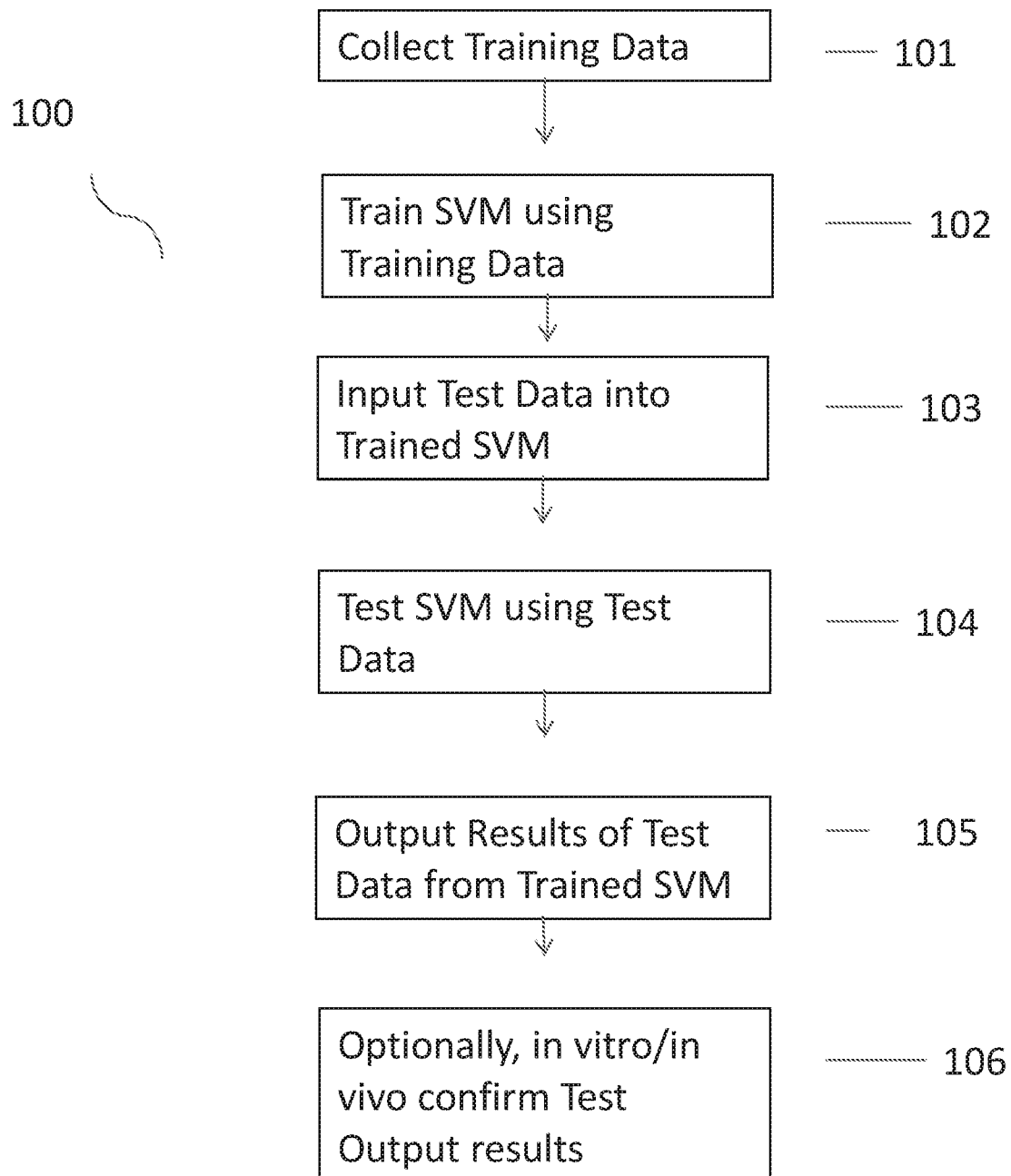
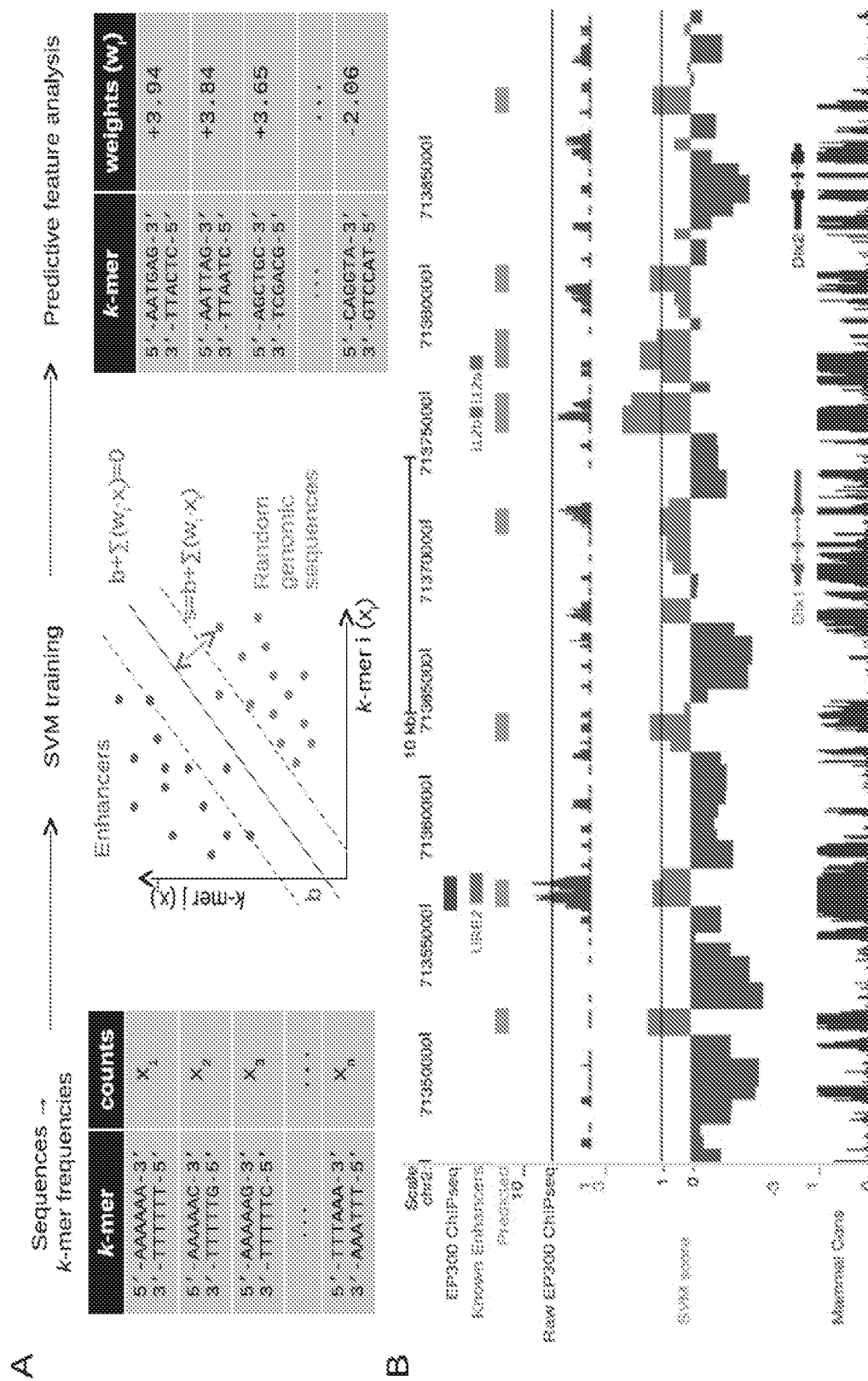


Fig. 1



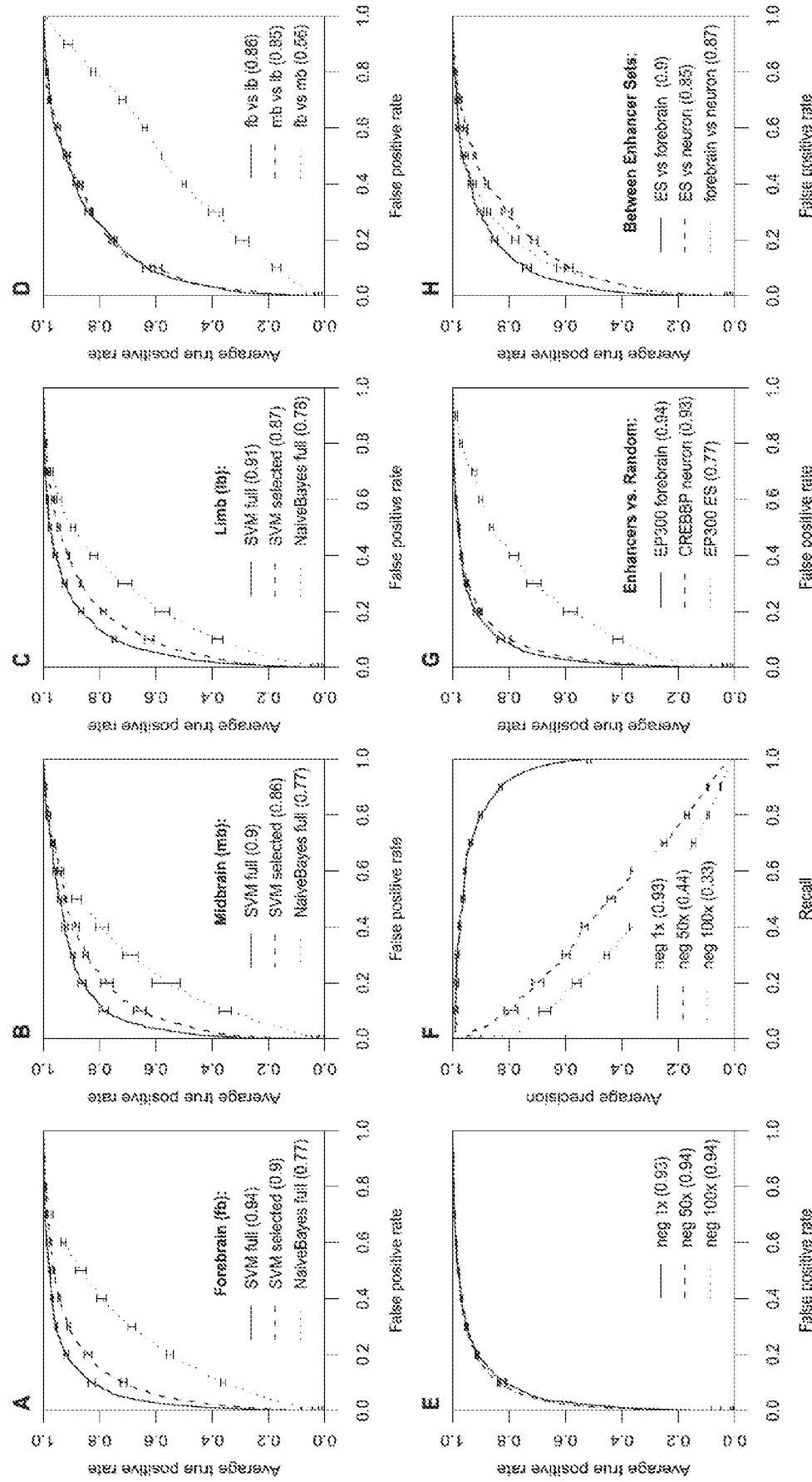


Fig. 3

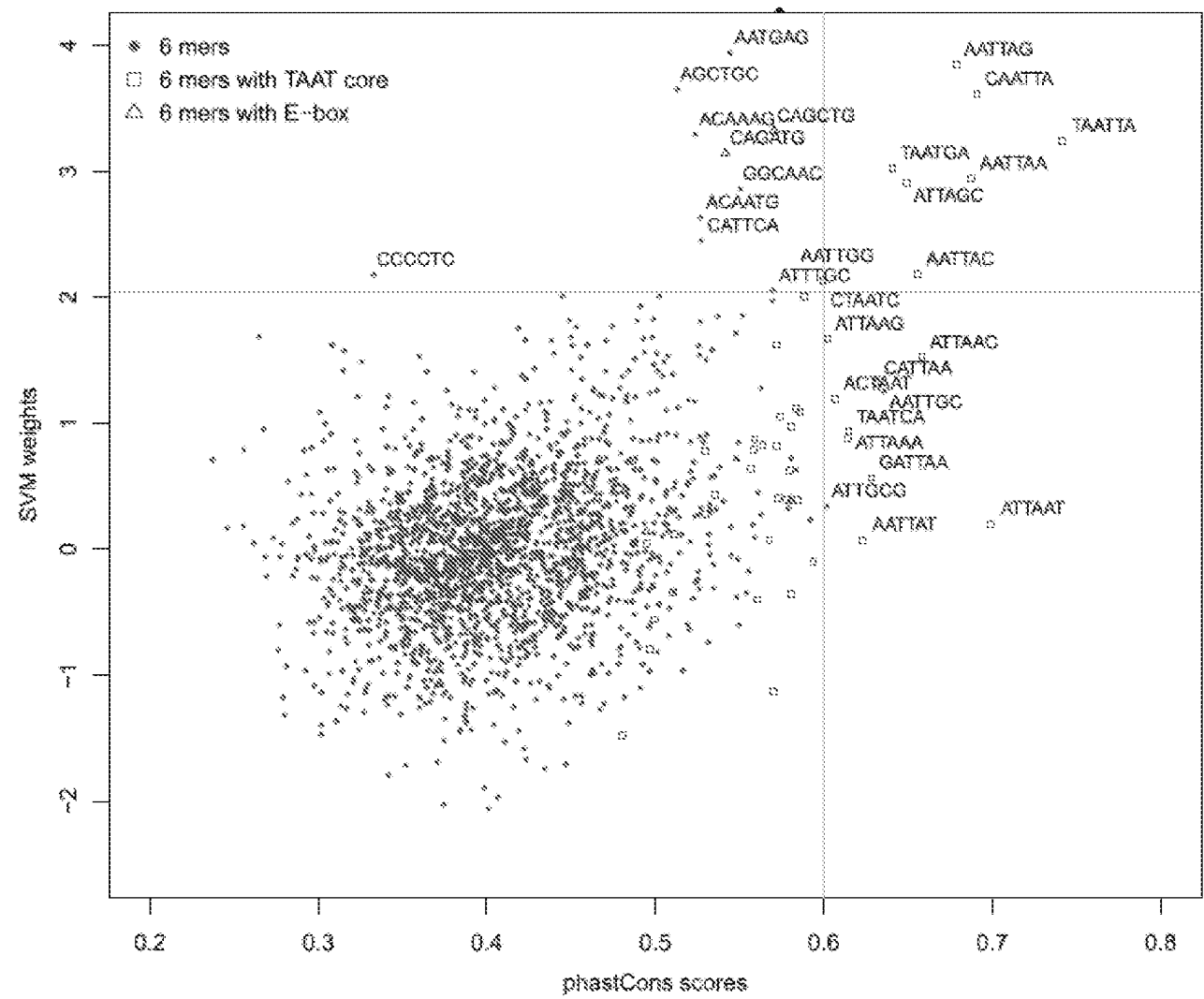


Fig. 4

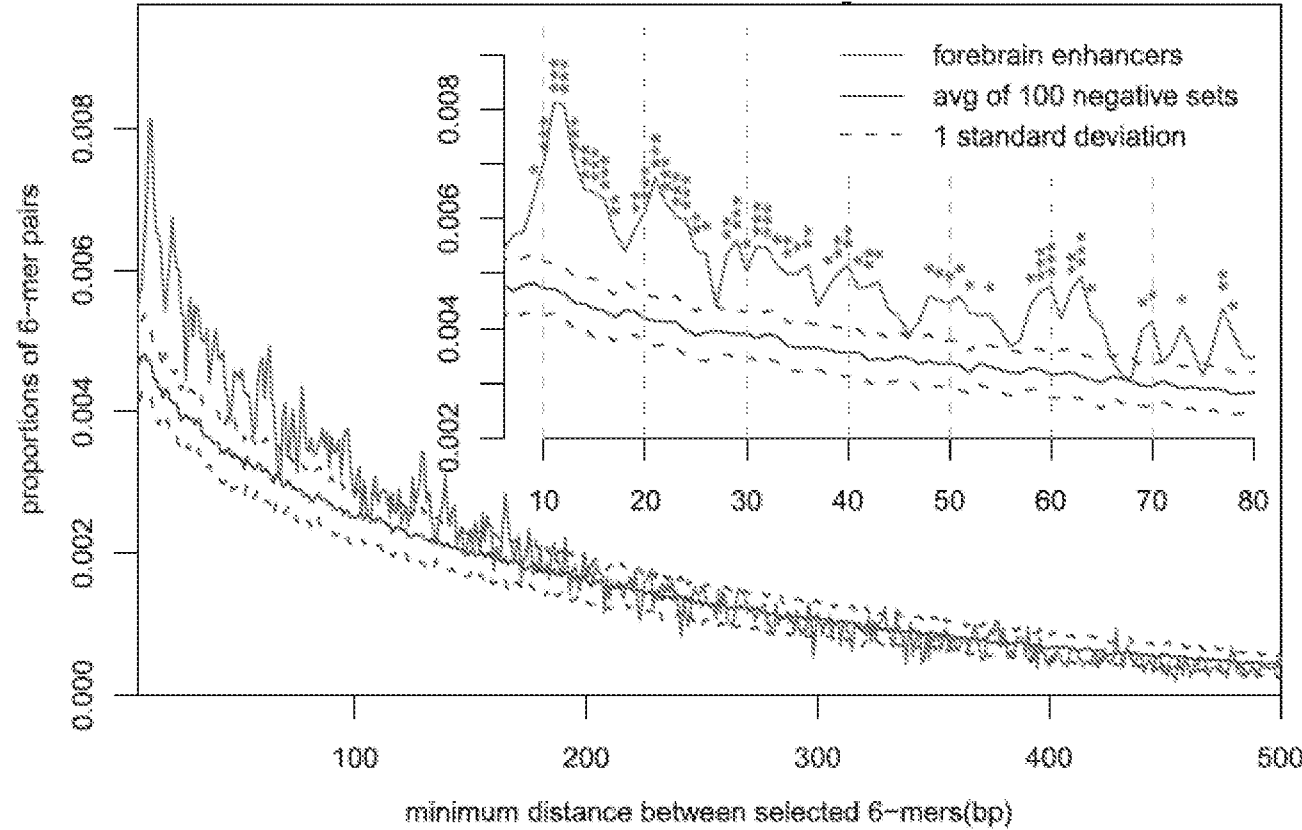


Fig. 5

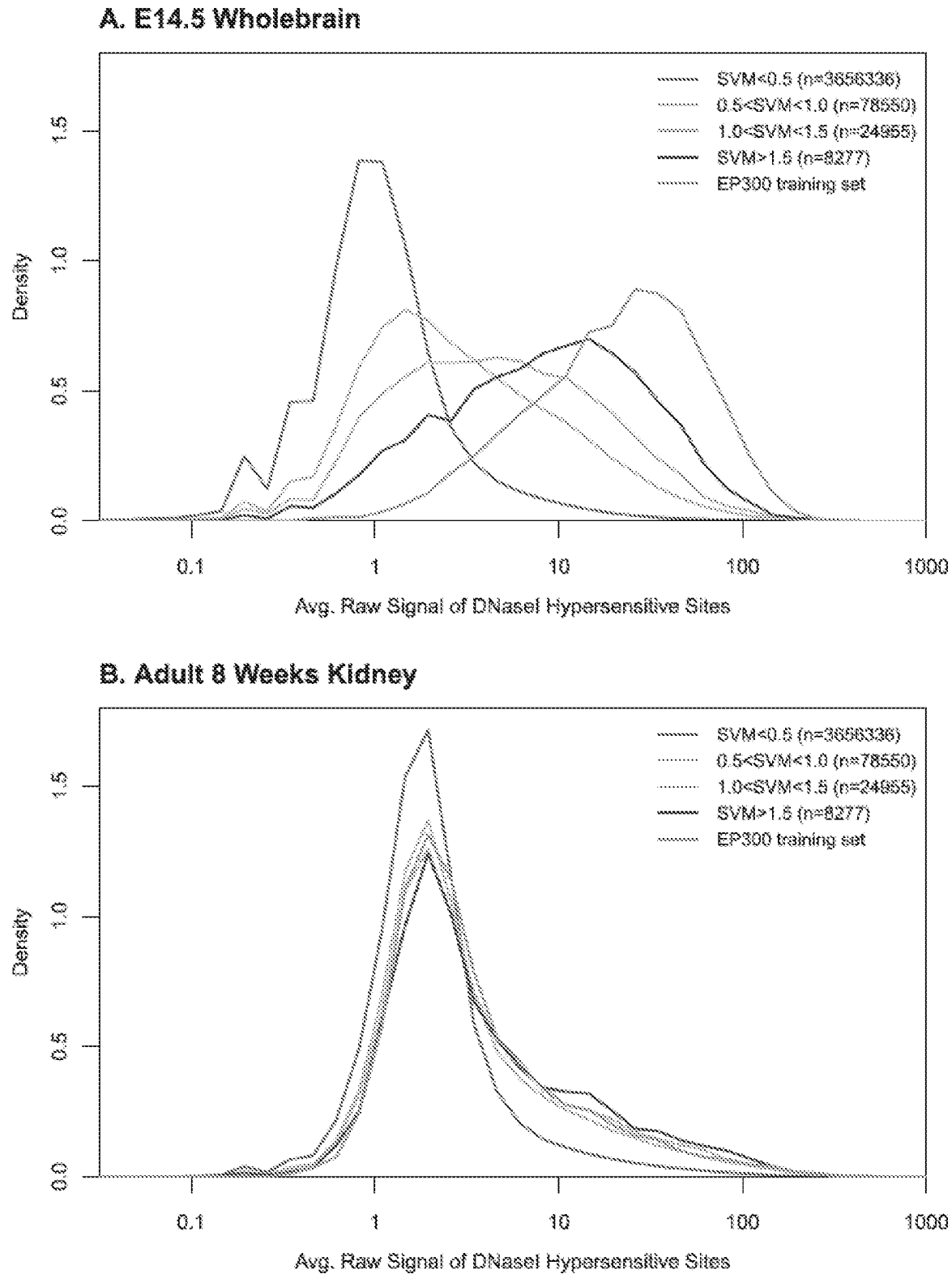


Fig. 6

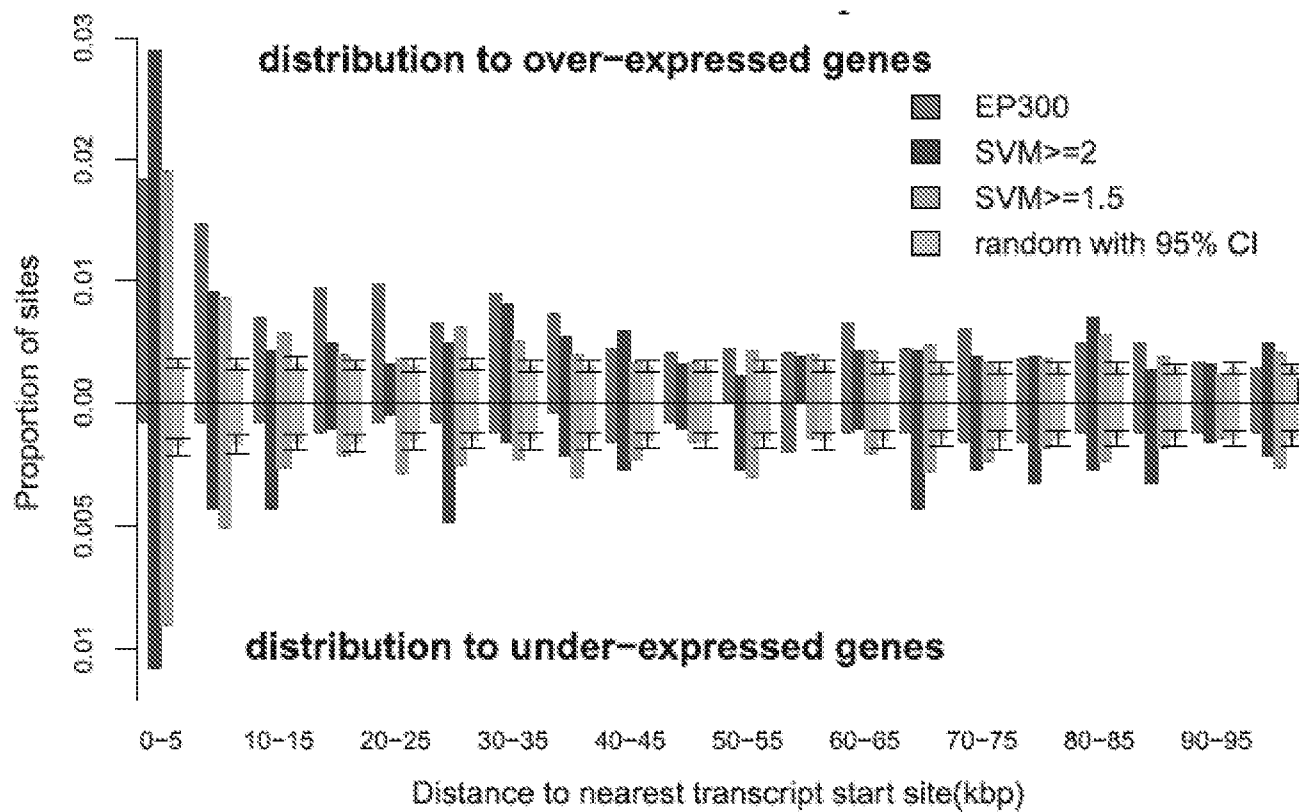


Fig. 7

Table 1. Predictive 6-mers of EP300 forebrain

(A) Fifteen 6-mers with the largest positive SVM weights				
6-mers	Reverse complement	SVM weight	Database family match	Top matched transcription factors (q-val < 0.1)
AATGAG	CTCATT	3.94	Homeodomain	POU6F1
AATTAG	CTAATT	3.85	Homeodomain	VSX2, PRRX2, EVX2, PDX1, GBX2
AGCTGC	GCAGCT	3.65	HLH	NHLH1, HEN1, ASCL2, REPIN1, TCF3
CAATTA	TAATTG	3.62	Homeodomain	BARHL2, PRRX2, NKX2-S, NKX6-1, BARHL1
CAGCTG	CAGCTG	3.32	HLH	NHLH1, HEN1, REPIN1, ASCL2, MYOD1, TCF3
ACAAAG	CTTTGT	3.29	SOX	SOX4, SOX11, SOX10, HNF4A
TAATTA	TAATTA	3.24	Homeodomain	OTP, PROP1, HOXA, ALX1, LHX3
CAGATG	CATCTG	3.15	HLH	ZFP238, TAL1:TCF3, TAL1:TCF4, TCF3
TAATGA	TCATTA	3.03	Homeodomain	POU6F1, POU4F3, LHX3, HOXC9, NKX6-3
AATTAA	TTAATT	2.94	Homeodomain	LHX3, OTP, PRRX2, PROP1, LHX5
ATTAGC	GCTAAT	2.90	Homeodomain	VSX2, POU3F2, EVX2, PITX3, LHX8
GGCAAC	GTTGCC	2.86	-----	-----
ACAATG	CATTGT	2.63	SOX	SOX17, SOX9, SOX5, SOX10, SOX30
CATTCA	TGAATG	2.45	SOX	HBP1
AATTAC	GTAATT	2.18	Homeodomain	PRRX2, HOXA6, HOXA1, HOXC8, DLX1
(B) Five 6-mers with the largest negative SVM weights				
AGGTAG	CTACCT	-1.79	--	--
AAGTCA	TGACTT	-1.89	--	--
AGGTGA	TCACCT	-1.97	Zinc-finger	ZEB1
ACCTGG	CCAGGT	-2.03	Zinc-finger	ZEB1, TCF3
CAGGTA	TACCTG	-2.06	Zinc-finger	ZEB1

Fig. 8**Table 2.** Precision and sensitivity of detecting DNase I hypersensitive enhancers

		True positives	False positives	Precision	Sensitivity
		DNase I > 10	DNase I < 2	TP/(TP+FP)	1-FP/N
Predicted positives	SVM > 1.5	3892	1330	74.5%	0.9996
	SVM > 1.0	11,081	8612	56.3%	0.997
Predicted negatives	SVM < 1.0	98,590	3,086,512	3.5%	
		P = 109,671	N = 3,095,124		

Fig. 9

(A) SVMs and Naïve Bayes Classifiers

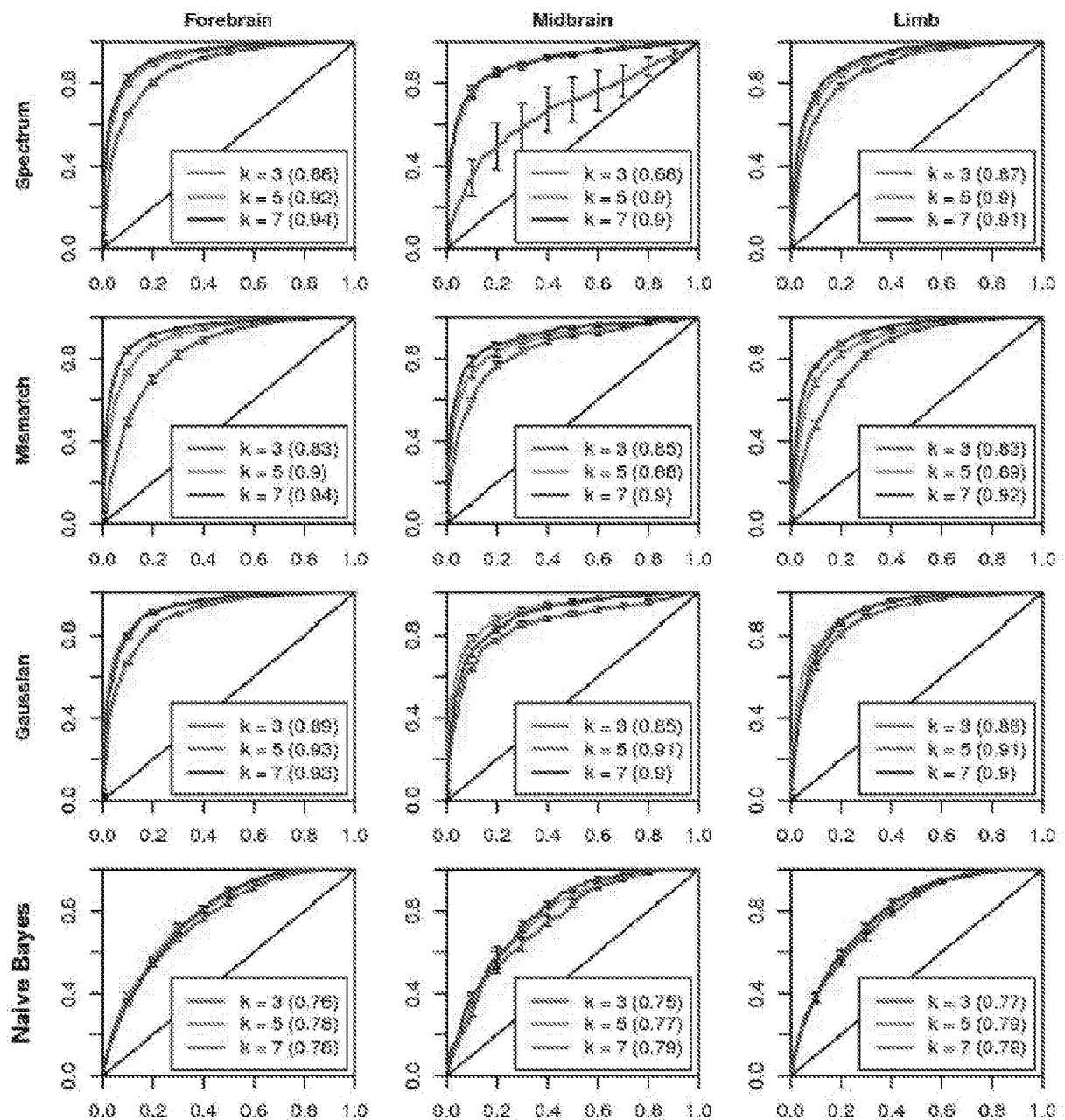


Fig. 10A

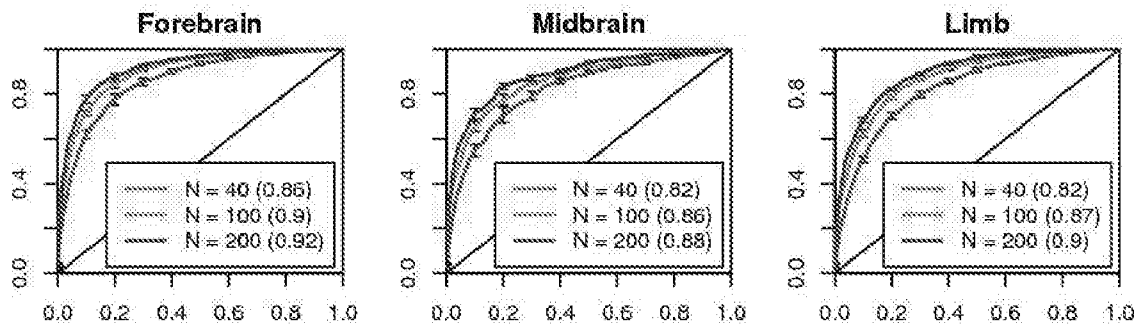


Fig. 10B

(a) Sequence Lengths

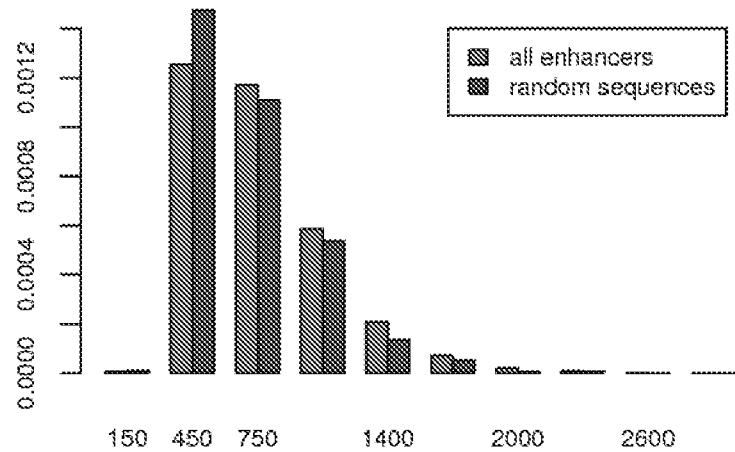


Fig. 11A

(b) Repeat Fractions

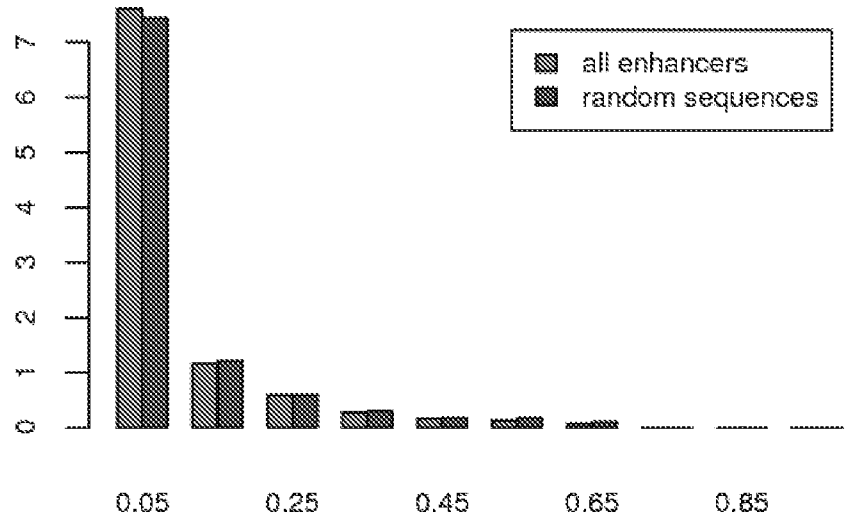


Fig. 11B

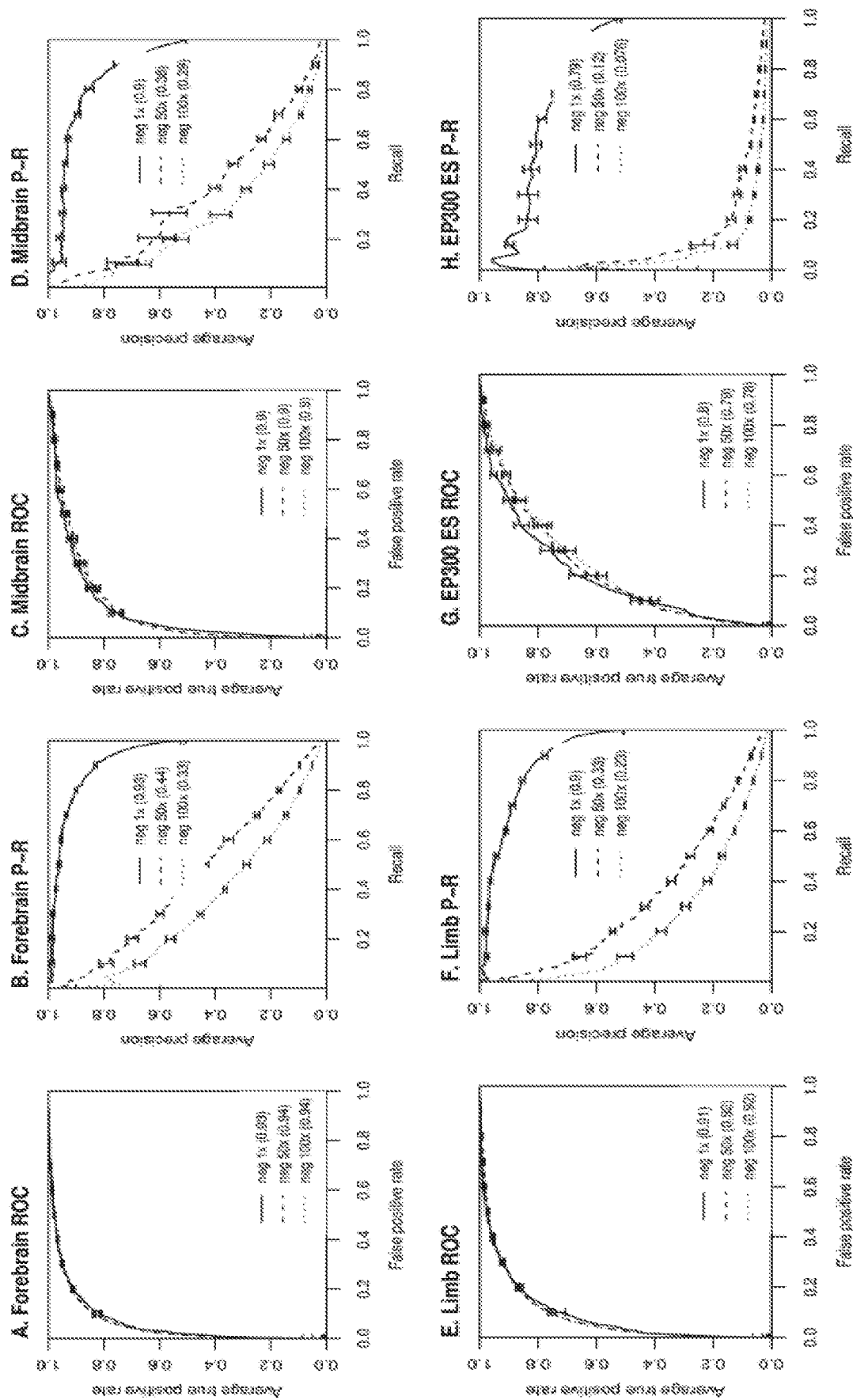


Fig. 12

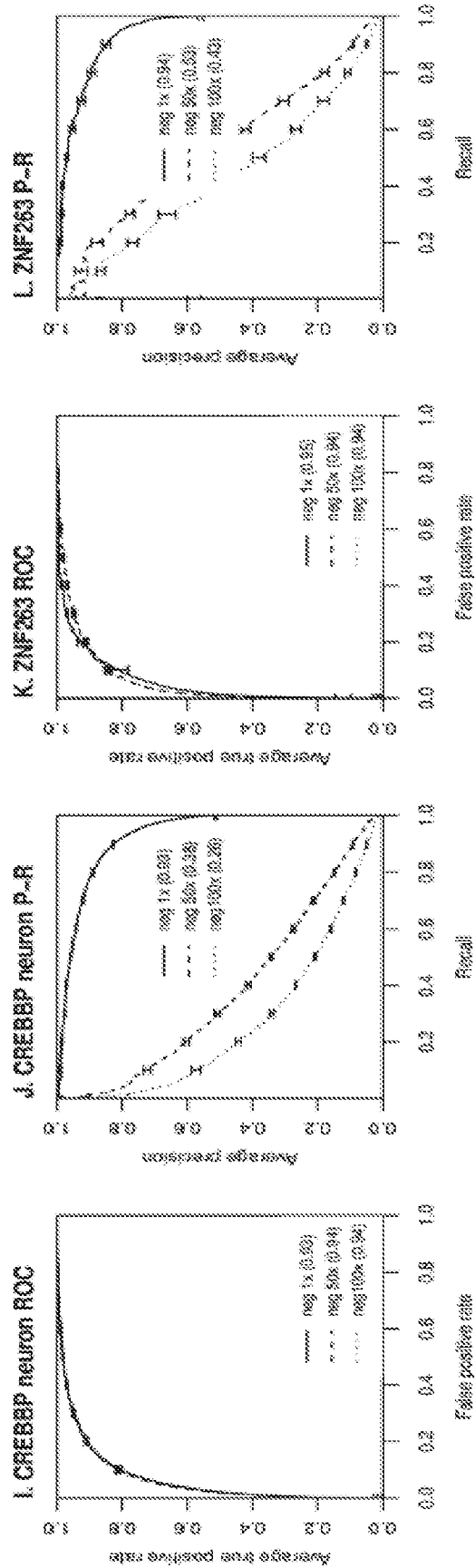
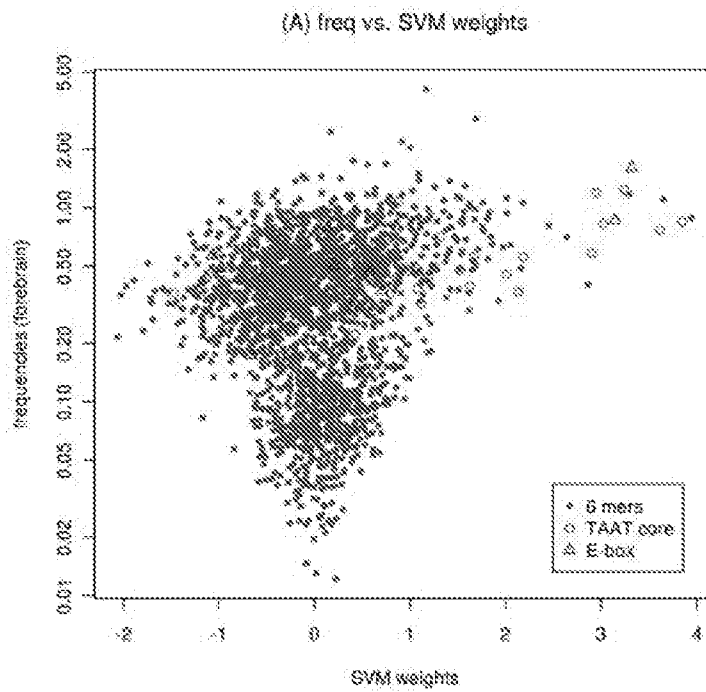
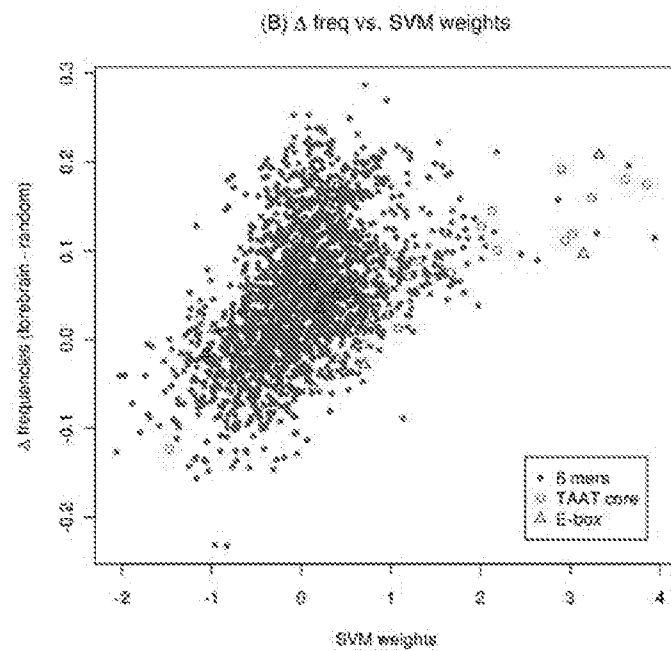


Fig. 12 continued

**Fig. 13A****Fig. 13B**

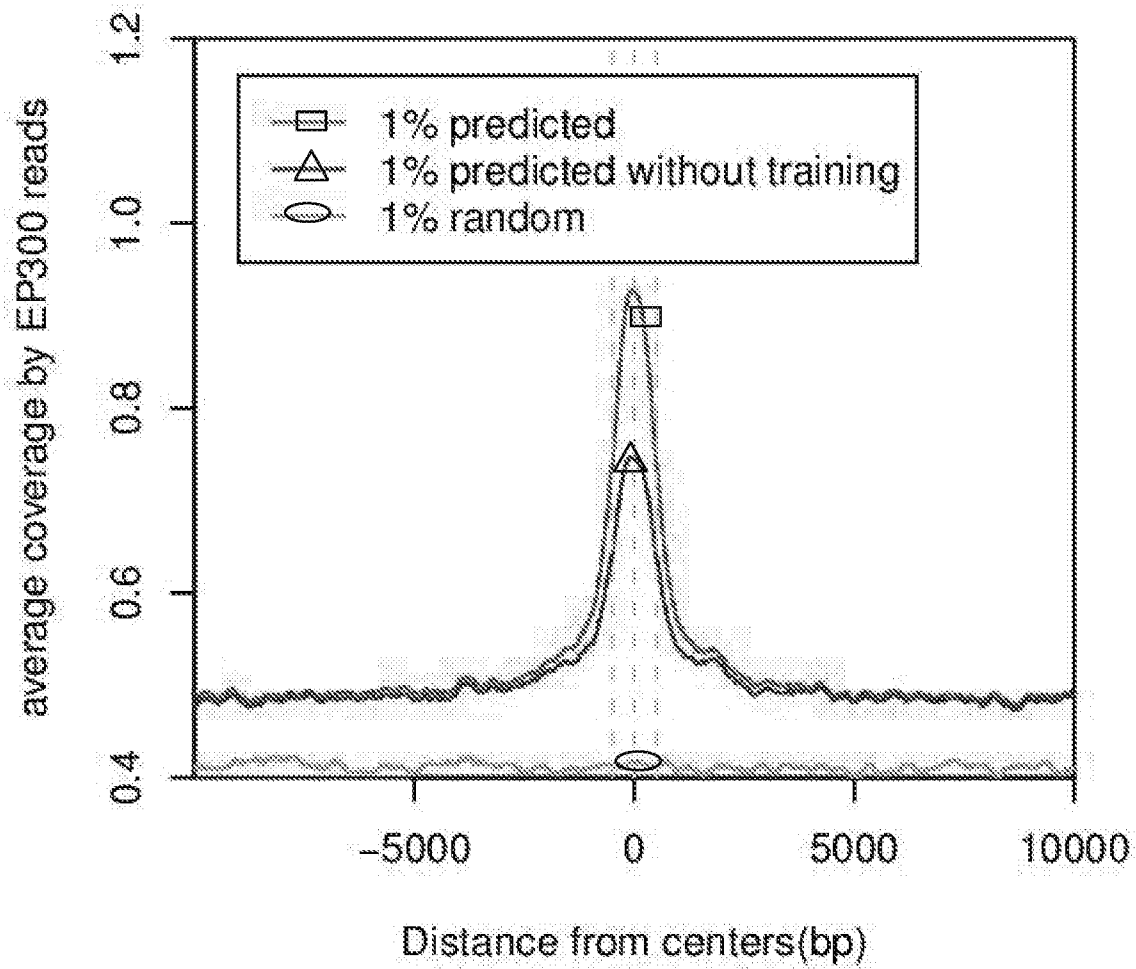


Fig. 14

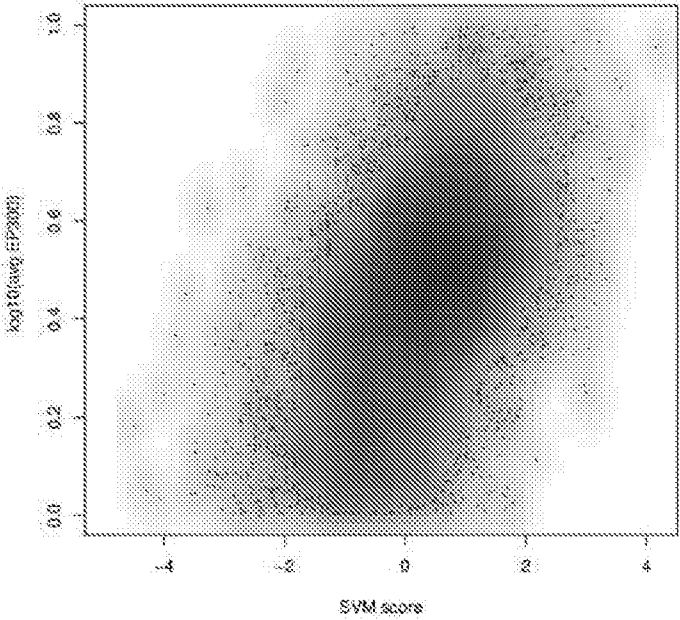


Fig. 15A

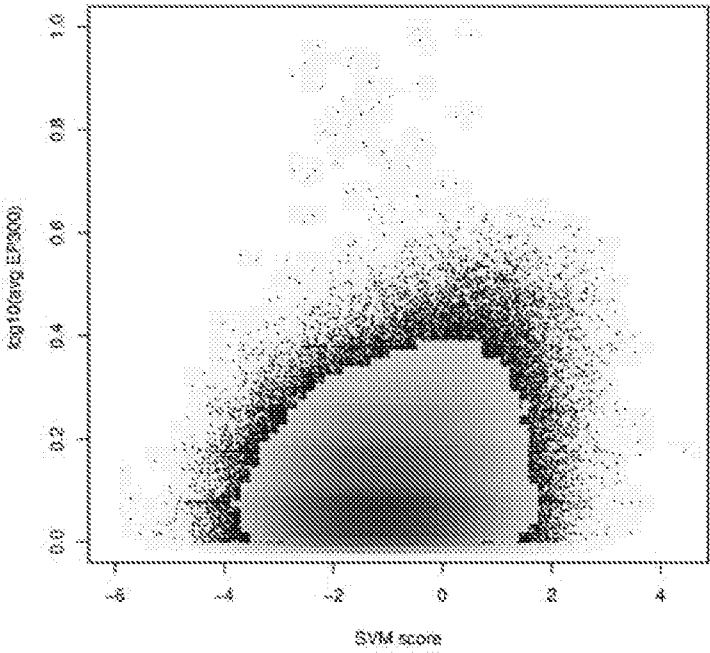
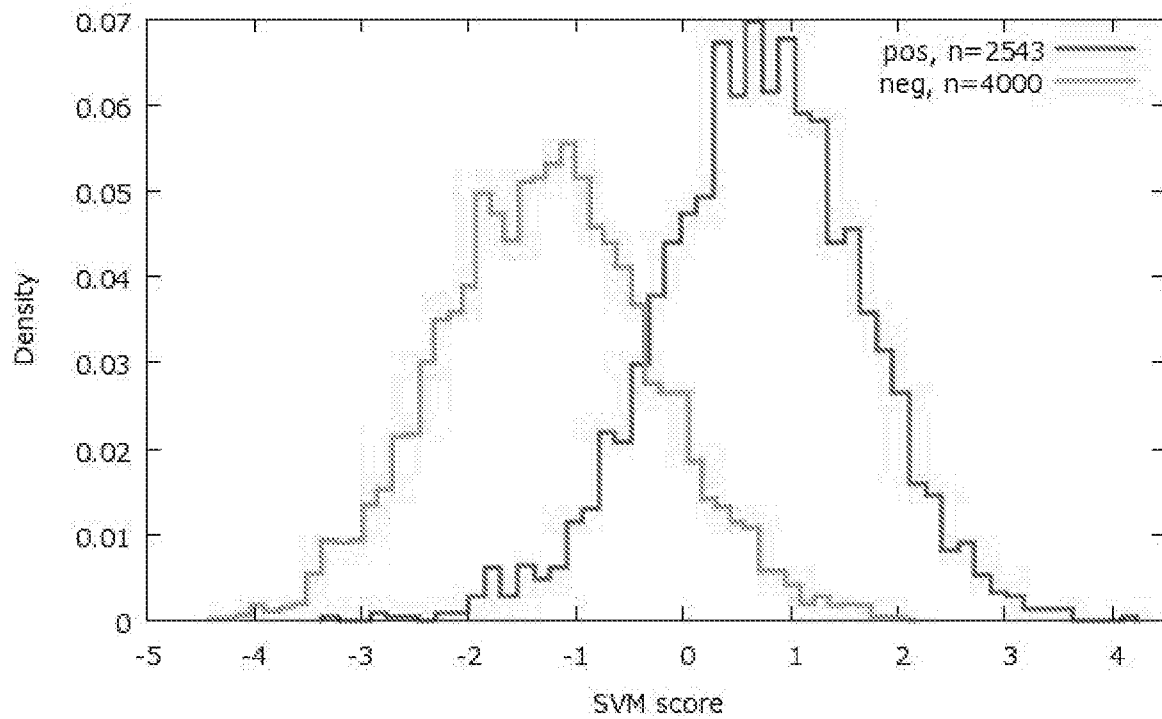
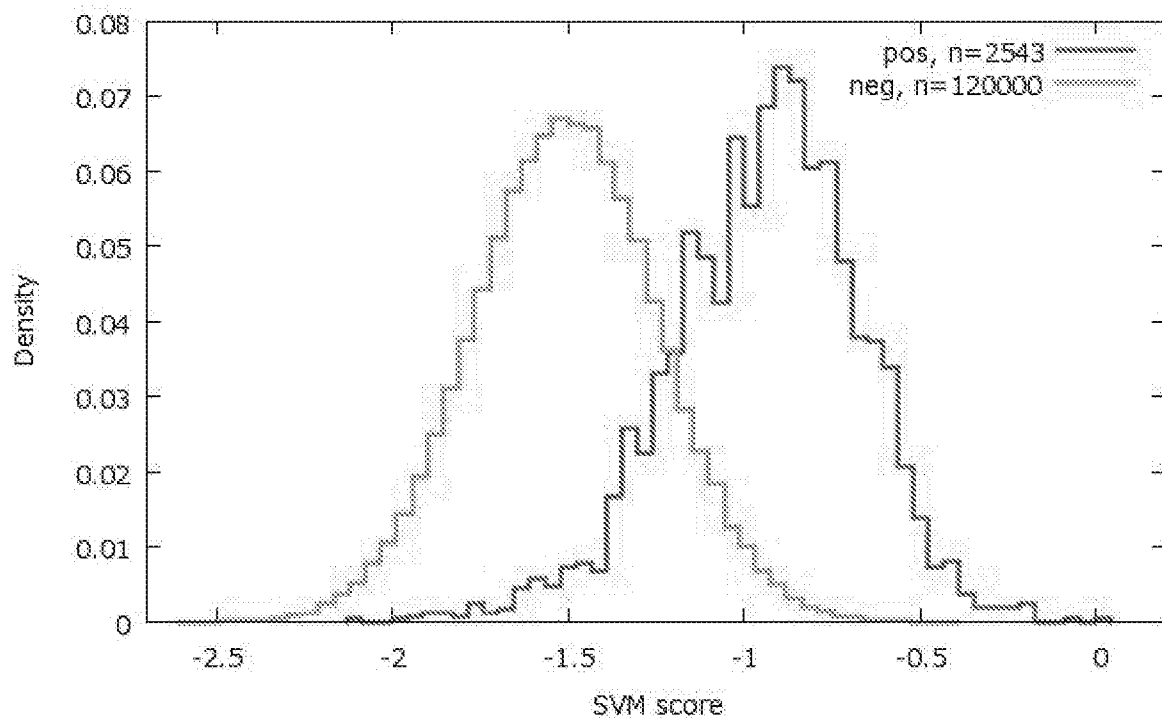


Fig. 15B

**Fig. 16A****Fig. 16B**

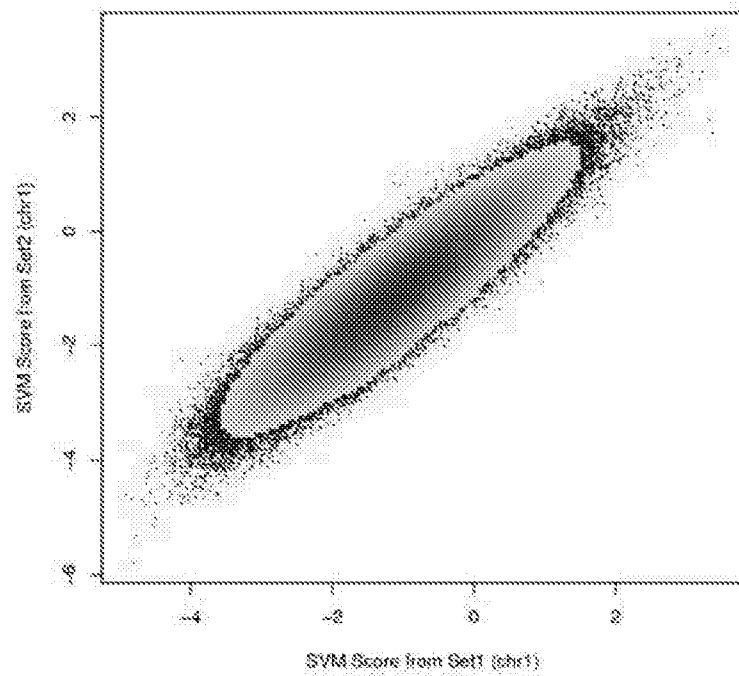


Fig. 17

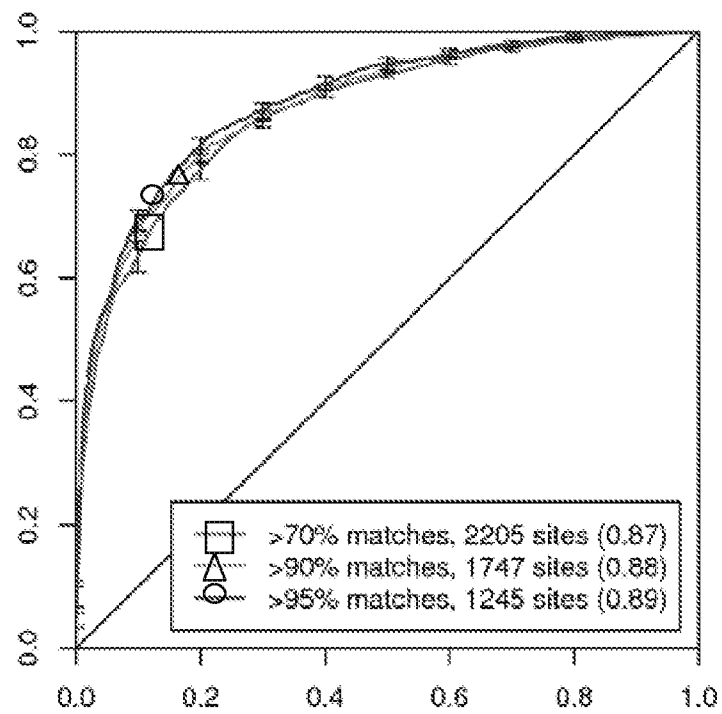


Fig. 18

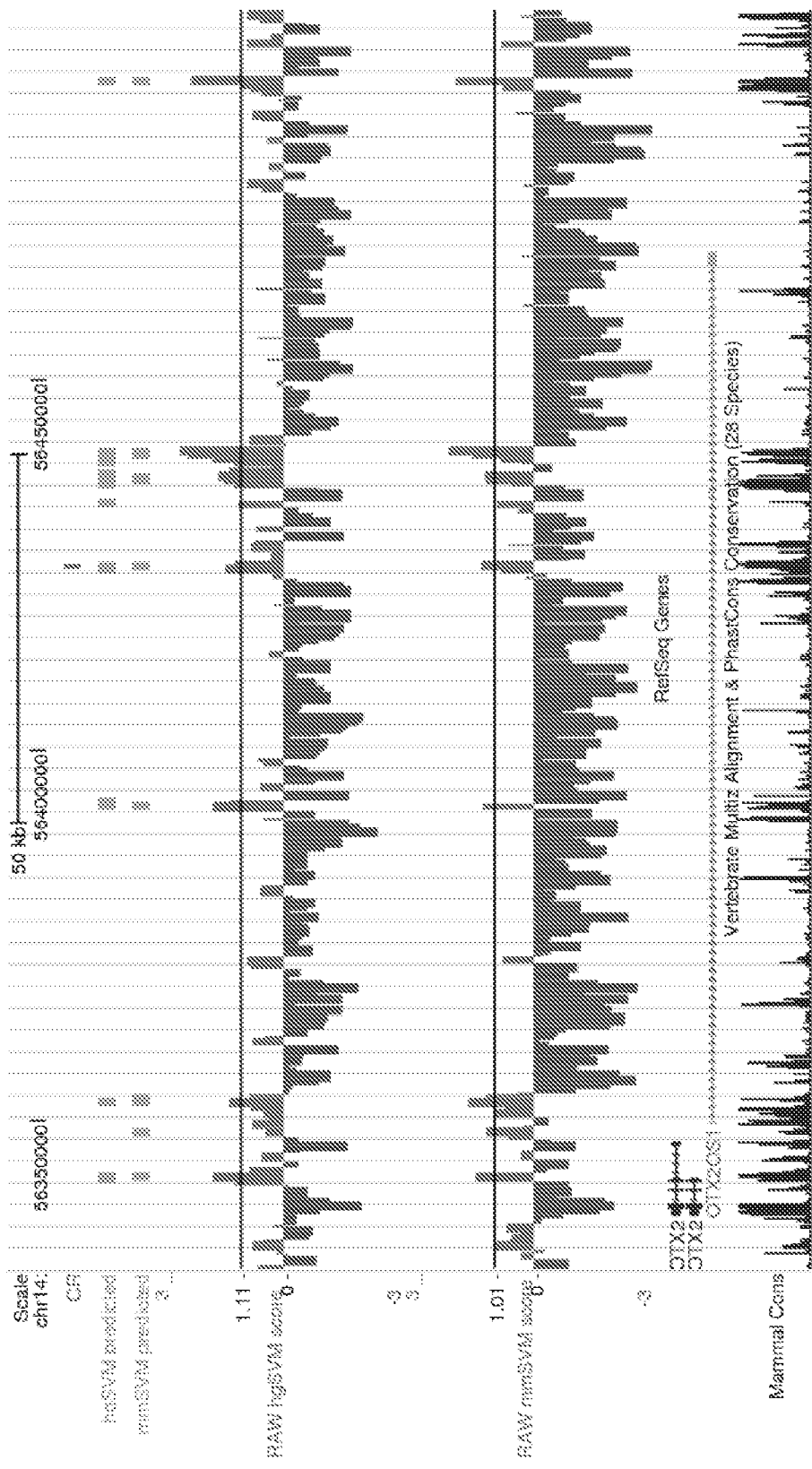


Fig. 19

Table 3

	SOX2	OCT4	weights
SOX2-OCT4- NANOG	CATTGTYATGCAAAT (SEQ ID NO. 2)		
SOX2:	CATTGT.....		1.45
SOX2:	.ATTGTY.....		1.19 or 0.77
SOX2:	..TTGTYA.....		0.85 or 0.32
OCT4:		TATGCA...	1.0
OCT4:		ATGCAA..	0.71
OCT4:	TGCAAA.		0.58
OCT4:	GCAAAT		0.47

Fig. 20

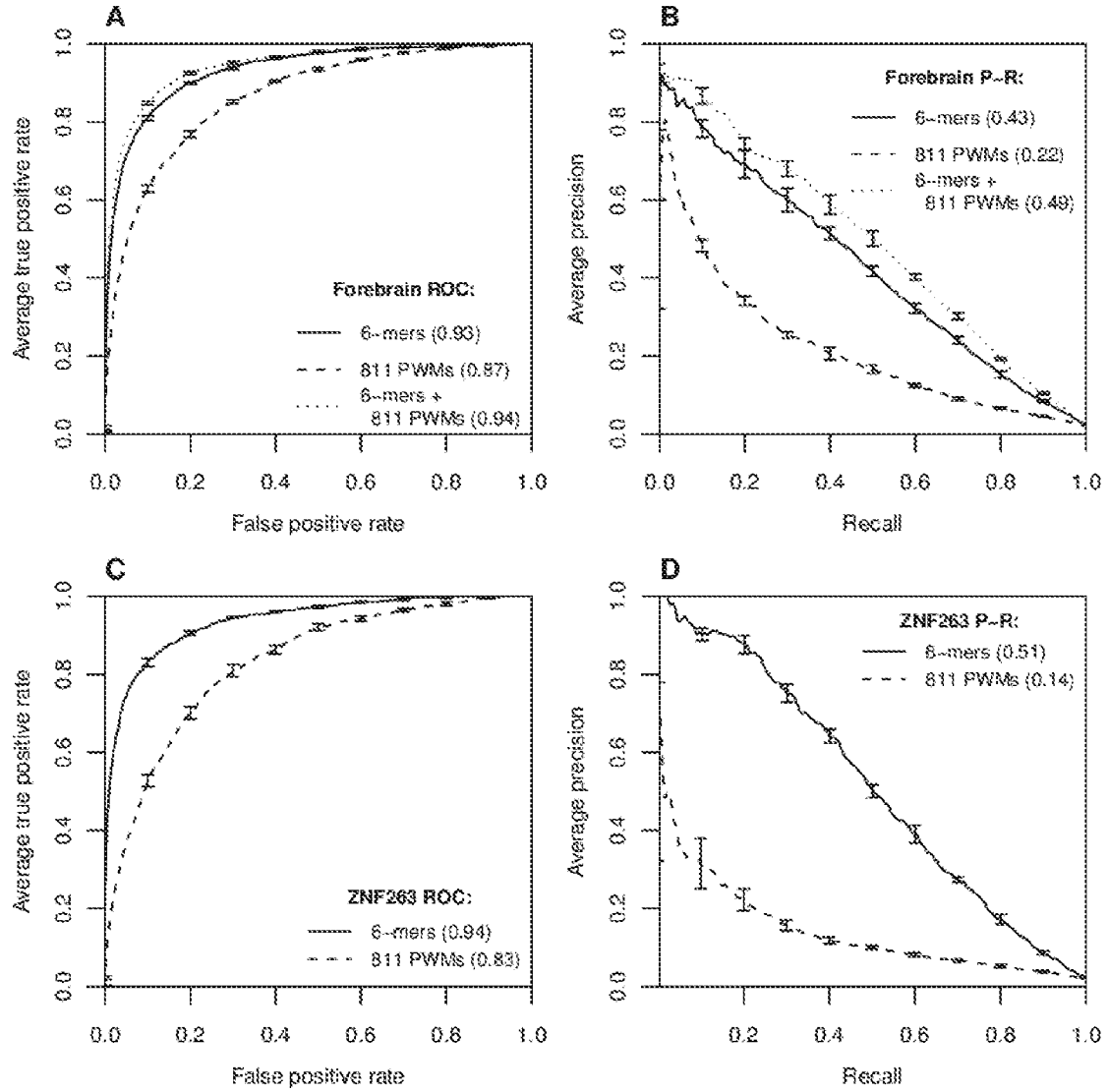


Fig. 21

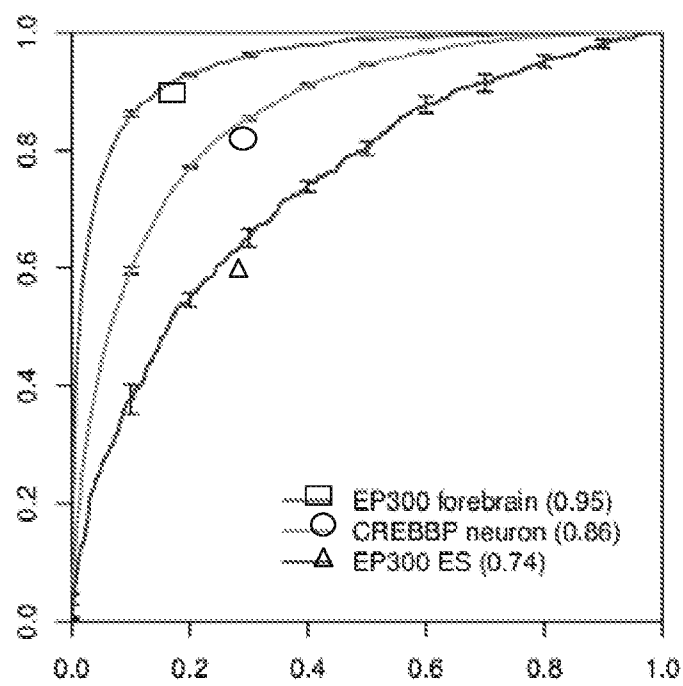


Fig. 22

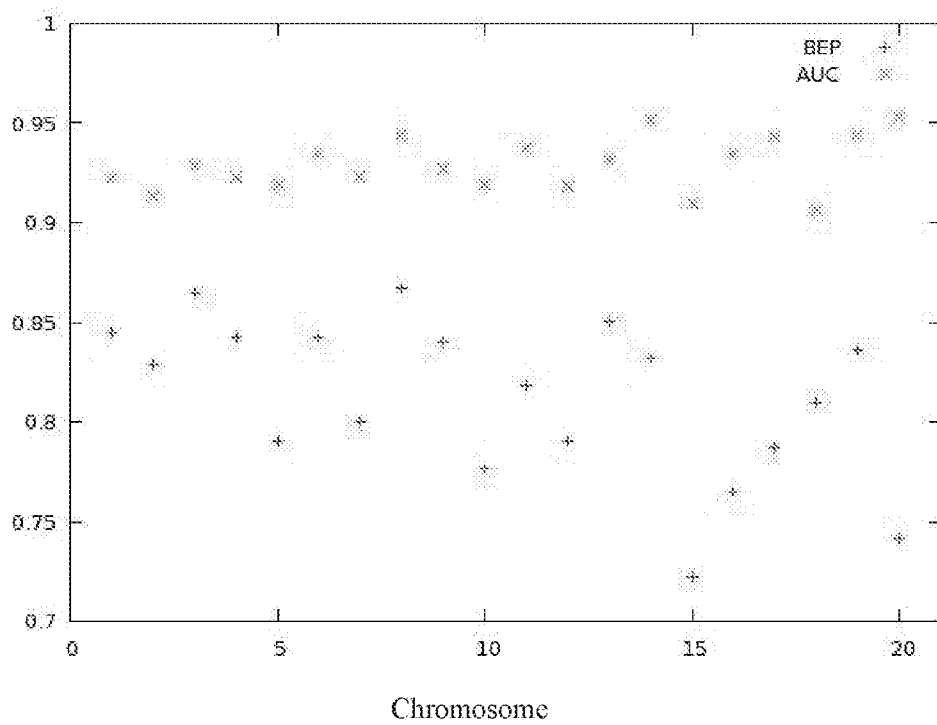


Fig. 23

Table 4: Outline of Analysis

Classification Experiments					
Experiment	Positive	Negative	Neg. set size	Applied Methods	Figures
Visel EP300 vs random	EP300 fb, mb, and lb	random genomic	4000	sk-SVM, msk-SVM, G-SVM, NB, L1-LR*, and KIRMES*	Fig 3, Fig 10A/B
Visel EP300 vs each other	EP300 fb,	EP300 mb	-	sk-SVM	
	EP300 fb	EP300 lb			
	EP300 mb	EP300 lb			
Negative set size scaling	EP300 fb, mb lb, EP300 ES, CREBBP neuron, ZNF263	random genomic	1x, 50x, 100x	sk-SVM	Fig 12A-L
Human	EP300 fb human orthologs	random human genomic	1x	sk-SVM	Fig 18
Other EP300/CREBBP datasets vs random	EP300 ES	random genomic	10x	sk-SVM	Fig 8
	CREBBP neuron	random genomic	1x	sk-SVM, WDS	
Other datasets vs each other	fb	EP300 ES	-	sk-SVM	
	fb	CREBBP neuron			
	EP300 ES	CREBBP neuron			
TF ChIP-seq	ZNF263	random genomic	50x	sk-SVM	Fig 21
<i>k</i> -mers vs PWM	EP300 fb, ZNF263	random genomic	4000, 50x	811 PWM SVM	Fig 21
Assessment Methodologies					
Similarity between predictive <i>k</i> -mers and known TFBS					Table 2
Conservation of predictive features between mouse and human					Fig 4
Spatial constraints between predictive features and relevant genes					Fig 5,7
DNaseI hypersensitivity of genome-wide enhancer predictions					Fig 6

* only applies to EP300 fb vs random genomic; sk-SVM, spectrum kernel; msk-SVM, mismatch spectrum kernel; G-SVM, Gaussian kernel; NB, Naïve Bayes; L1-LR, logistic regression; WDS, weighted degree kernel with shifts.

Fig. 24

Table 5: Overlap Between Top SVM Scoring Regions Determined by Two Separately Trained SVMs

Set1	Set2	Number of sites only in set1 (a)	Number of sites only in set2	Number of sites in both sets (b)	% Overlap (100*b/(a+b))
0.5%	0.5%	9815	9815	16575	62.8
1.0%	1.0%	18711	18711	34069	64.5
1.0%	2.0%	8156	60938	44624	84.5
1.0%	3.0%	4119	109679	48661	92.2
1.0%	5.0%	1313	212432	51467	97.5

Fig. 25**Table 6:** Human Enhancer Prediction Using a Mouse vs. a Human SVM

Set1 (mouse)	Set2 (human)	Number of sites only in Set1 (a)	Number of sites only in Set2	Number of sites in both sets (b)	Overlap (100*b/(a+b))
0.5%	0.5%	19147	19147	9531	33.2
1.0%	1.0%	36819	36818	20537	35.8
1.0%	2.0%	28489	85843	28867	50.3
1.0%	3.0%	23352	138060	34004	59.3
1.0%	5.0%	16825	246242	40531	70.7

Fig. 26

Table 7: Overlap Between EP300 and CREBBP Binding in Different Data Sets

Regions	Number of observed peaks in the regions	Number of total peaks	Probability of peaks being in a region	Expected number of peaks in the regions	<i>p</i> -value
forebrain	408	11847	0.021	249.64	< 2.2e-16
midbrain	72	11847	0.004	50.18	0.002143
limb	67	11847	0.015	179.27	1

A

Regions	Number of observed peaks in the regions	Number of total peaks	Probability of peaks being in a region	Expected number of peaks in the regions	<i>p</i> -value
forebrain	11	524	0.021	11.04	0.5464
midbrain	3	524	0.004	2.22	0.3826
limb	1	524	0.015	7.93	0.9997

B

Fig. 27

Table 8: Predictive 6-mers of CREBBP Neuron

6-mers	Reverse Complement	SVM weight	Database Family Match	Top Matched Transcription Factors (q-val <0.1)
GACTCA	TGAGTC	5.68	Leucine Zipper	BACH1, NFE2, BACH2, JUNDM2, AP1
CAGATG	CATCTG	5.25	HLH	ZNF238, TAL1-TCF3, TAL1-TCF4, TCF3
GAGTCA	TGAGTC	5.19	Leucine Zipper	BACH1, BACH2, NFE2, JUNDM2, AP1
ACGTCA	TGACGT	5.05	Leucine Zipper	ATF6, CREB, ATF1
TGCCAA	TTGSCA	4.80	Nuclear Factor I	HIC1, NFIC, NF1
AGATGG	CCATCT	4.39	HLH	TAL1-TCF3, TAL1-TCF4, YY1, TCF3
CATATG	CATATG	4.13	-	-
AATTAG	CTAATT	4.02	Homeodomain	V5X2, PRRX2, EVX2, PDX1, GBX2
GGCAAC	GTTGCC	3.63	-	-
CTGSCA	TGCCAG	3.47	Nuclear Factor I	HAND1-TCF3, HIC1, NFIC, TGIF1
TAATTA	TAATTA	3.42	Homeodomain	OTF, PRO1, HOXA, ALX1, LHX3
GATCA	TGATC	3.30	-	-
CGTCAC	GTCACG	3.18	Leucine Zipper	PAX3, CREB, ATF1, JUNDM2
GCCTCA	TGACGC	3.13	Leucine Zipper	CREB, ATF6, BACH1, BACH2
AATTAC	GTAATT	3.10	Homeodomain	PRRX2, HOXA6, HOXA1, HOXC3, DLX1

A

6-mers	Reverse Complement	SVM weight	Database Family Match	Top Matched Transcription Factors (q-val <0.1)
GGTCRA	TTGACC	-3.58	Nuclear Receptor	PPARG, RORA2, HNF4A, RORA1, ESRR
CTGACC	GGTCAG	-3.98	Nuclear Receptor	RORA2, RORA1, NR1F2, ESRR, PPARG
AGGTGA	TCACCT	-3.99	Zinc-finger	ZEB1
CAGGTA	TACCTG	-4.08	Zinc-finger	ZEB1
GGGTCA	TGACCC	-4.64	Nuclear Receptor	NR1F2, ESRR, HNF4A, RXRA, PPARG

B

Fig. 28

Table 9: Predictive 6-mers of Embryonic Stem Cells

6-mers	Reverse Complement	SVM weight	Database Family Match	Top Matched Transcription Factors (q-val <0.1)
ACAATG	CATTGT	1.43	SOX	SOX17, SOX9, SOX3, SOX10, SOX30
ATIGTC	GACAT	1.19	SOX	SOX17
ACAAGG	CTTGT	1.06	SOX	SOX4, SOX11, SOX10, HNF4
TATGCA	TGCATA	1.00	Homeodomain	POU2F1, POU3F1, POU2F1, POU2F1
GAGCTA	TAGCTC	0.95	-	-
CABAAG	CTTTTG	0.90	-	-
AGHICA	TGACCT	0.89	Nuclear Receptor	RORA1, PPARG, RORA2, ESRRB, RAR
AAAGCC	GGCTTT	0.89	-	-
AATTC	GGAAAT	0.88	-	-
AAGGTC	GACCTT	0.88	Nuclear Receptor	PPARG, ESRRB, ESRA, RAR, NR1F2
TCTACA	TGTAGA	0.85	-	-
TAACAA	TTGTTA	0.85	SOX	SOX5
CCGGAA	TTCCGG	0.84	ETS	ELK4, GABPA, NRF2, STAT3, ELK1
GGTGAC	GTCACC	0.82	Leucine Zipper	SREBP, CREB, AP1, ATF, RAR
CATICA	TGAATG	0.79	SOX	HBP1

A

6-mers	Reverse Complement	SVM weight	Database Family Match	Top Matched Transcription Factors (q-val <0.1)
AACATG	CATGTT	-0.75	-	-
GCTAGA	TCIAGC	-0.76	-	-
CTGATA	TATCAG	-0.82	-	-
AATAAA	TTTAAAT	-0.83	Homeodomain	HOXD13, FOXO1, HOXB13
ACAAAT	ATTTGT	-0.85	-	-

B

Fig. 29

Table 10: Comparison of Predictive k-mers From the Different Data Sets

ES (+) vs forebrain (-)		ES (+) vs neuron (-)		forebrain (+) vs. neuron (-)	
AGGTCA	2.83917	AGGTCA	1.38178	ACAAAG	3.20456
ACCTTG	2.34974	CAATAG	1.08993	AGCTGC	2.63237
AGGTGA	1.97262	ACCTTG	0.954087	TAATGA	2.4704
CCTTGA	1.77742	AAGGTC	0.853491	CAGCTG	2.4703
AAGGTG	1.76077	CCTTGA	0.760972	GAACAA	2.46156
CACACC	1.64343	GGTCAC	0.73022	AAAGGG	2.33583
GGTGGA	1.52651	COGGAG	0.709136	GGATTA	2.23093
CAGGTA	1.51114	ACCTGA	0.671658	ACAATG	2.22637
CACCTG	1.48868	CACCTG	0.666395	CAATTA	2.22224
CTGACC	1.45141	CAGGTA	0.639238	AATTAG	2.10941
ACCTGG	1.44535	TCTACA	0.628242	CAATGG	2.08748
AGGTAA	1.42999	GGTCAA	0.627209	ATTAGC	2.05797
GAGTCA	1.38264	ACACCC	0.624997	GGCCCC	2.01011
CTAGAA	1.32414	GAACCC	0.624728	ACAATA	1.89593
AGGAAG	1.30616	AGGTGA	0.619084	GAGGCC	1.88978

Fig. 30A

Table 10 continued

ES (+) vs forebrain (-)		ES (+) vs neuron (-)		forebrain (+) vs. neuron (-)	
CTGGCA	-1.36987	ACAGAT	-0.58506	ATTTCA	-1.49916
AGGGGG	-1.38789	ATGACG	-0.593	GTGCCA	-1.52618
AATTAG	-1.39198	AACATG	-0.605551	CTCATC	-1.53853
GCTGCC	-1.41468	ATGCCA	-0.629451	ACTCAT	-1.60758
CAGCTG	-1.49427	AATATG	-0.637472	ACGTCA	-1.67551
CAGATG	-1.5308	CTAAAA	-0.647749	AACATG	-1.69289
ACAAAG	-1.54907	TAATTA	-0.660014	AAATTA	-1.71518
TAATTA	-1.6466	GGCAAC	-0.660122	GTTCCA	-1.73856
AGCTGC	-1.65257	CAGATG	-0.670137	GACTCA	-1.77009
GGCAAC	-1.66172	ACGTCA	-0.726237	CTAAAA	-1.9048
CAATTA	-1.68426	TGCCAA	-0.773441	GAATCA	-1.90671
AATGAG	-1.68725	CTGGCA	-0.776692	CATCTC	-1.94972
AATTAA	-1.76872	ATGTCA	-0.817955	GATTCA	-2.34549
CACAAA	-1.80749	GAAATA	-0.824868	GAGTCA	-2.46636
TAATGA	-1.882	AAATAG	-0.833908	TGCCAA	-3.12478

Fig. 30B

Table 11: Predictive k-mers of Three Different Data Sets Using Common Random Negative Sequences

Forebrain		Neuron		ES (w=5)	
AATTAG	5.166	TGCCAA	6.09624	AGGTCA	4.11367
CAATTA	4.53013	GACTCA	5.72339	ACAATG	3.33138
CAGCTG	4.00637	GAGTCA	4.82838	ACCTTG	2.98221
AGCTGC	3.95227	CAGATG	4.59956	AAGGTC	2.81142
ACAAAG	3.91098	CATATG	4.39705	CAATAG	2.66733
TAATGA	3.82868	AGATGG	4.28309	ATTGTC	2.44483
ATTAGC	3.79013	CTGGCA	4.23722	AAGCCG	2.24401
AATGAG	3.70752	ACGTCA	4.17313	CCGGAG	2.20552
TAATTA	3.60443	AATTAG	4.01272	CCGCCA	2.11105
CAGATG	3.49841	TAATTA	3.74813	TCGGAA	2.08395
GGCAAC	3.30678	GGCAAC	3.51165	CGGCTA	2.00939
ACAATG	2.79336	AATTAC	3.49265	ACGCCT	1.99696
CATTCA	2.71148	GAATCA	3.33674	ACACCC	1.9699
TGCTAA	2.52389	CTTGGC	3.30323	AATCCC	1.9580
AATTGG	2.43143	GCCAAA	3.23247	AATTCC	1.95536

Fig. 31A

Table 11 continued

Forebrain		Neuron		ES (w=5)	
AGGTAA	-1.71708	AGGTCA	-2.30776	GTCATA	-1.69074
CCATGA	-1.73288	CGGTCA	-2.3817	ATGTCA	-1.71948
AGGATG	-1.73706	AAGGTG	-2.46491	CACTAC	-1.75958
ACCTGA	-1.89247	AAAGTT	-2.54574	AACATG	-1.78623
AGGTAG	-1.99038	GGTCAA	-2.65552	ATCCAA	-1.78928
AGGTGA	-2.03761	ATGACC	-2.66935	AAAAAG	-1.87093
CATCTA	-2.10399	TGACCA	-2.76447	GAAATA	-1.94475
AAGTCA	-2.1352	AGGTAA	-2.99852	GACCGA	-2.02932
ACCTGG	-2.39139	AGGTGA	-3.39747	ACTCGA	-2.12442
CAGGTA	-2.92076	CAGGTA	-4.02767	CAATTC	-2.19647

Fig. 31B

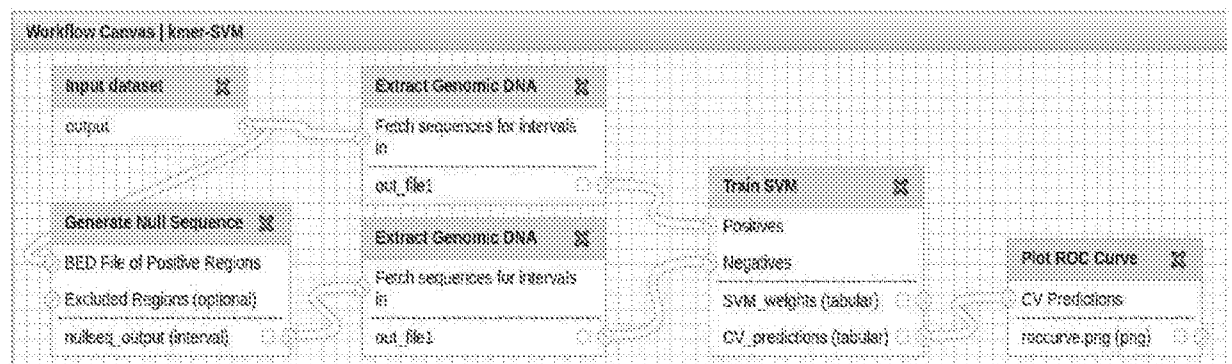


Fig. 32

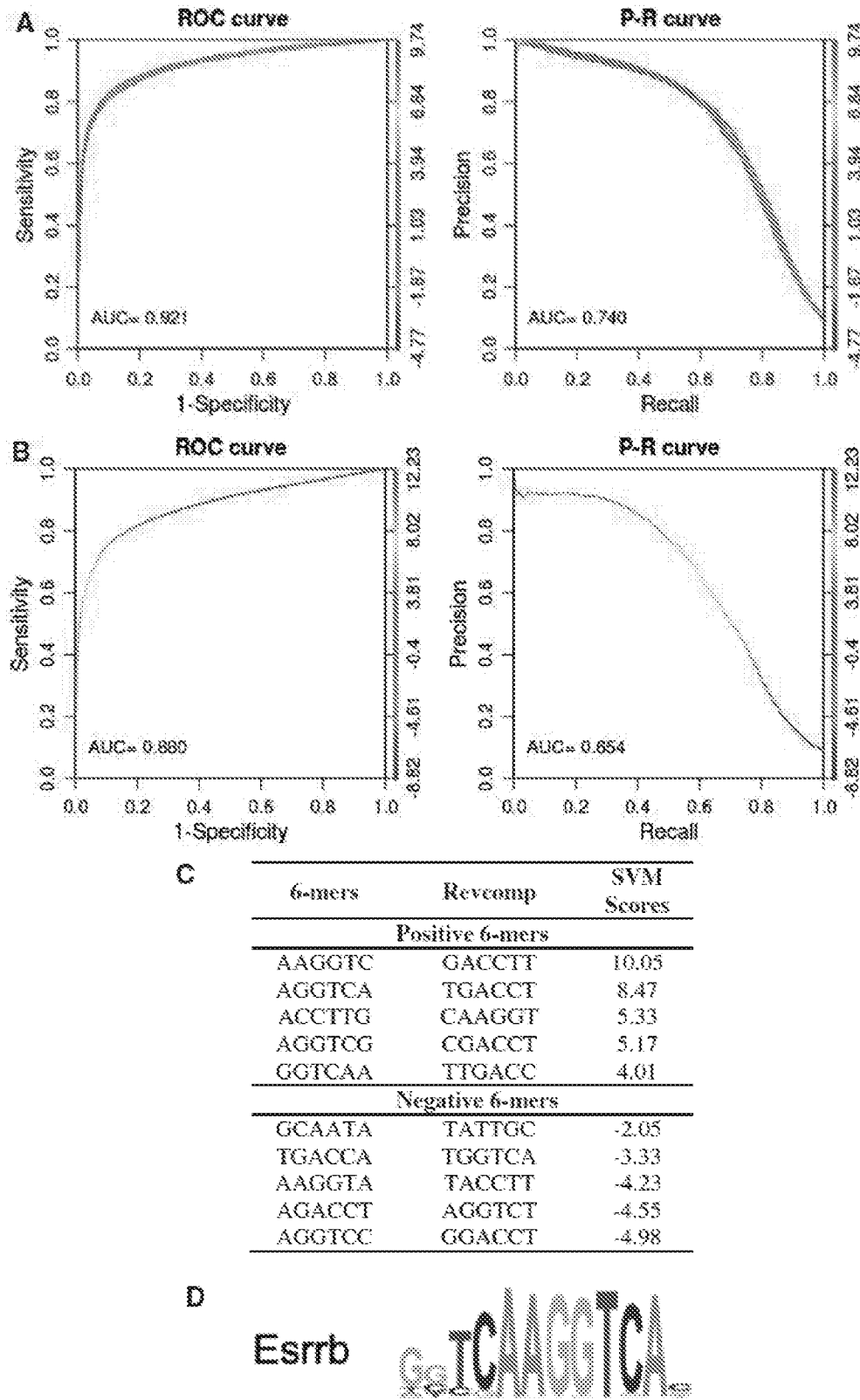


Fig. 33

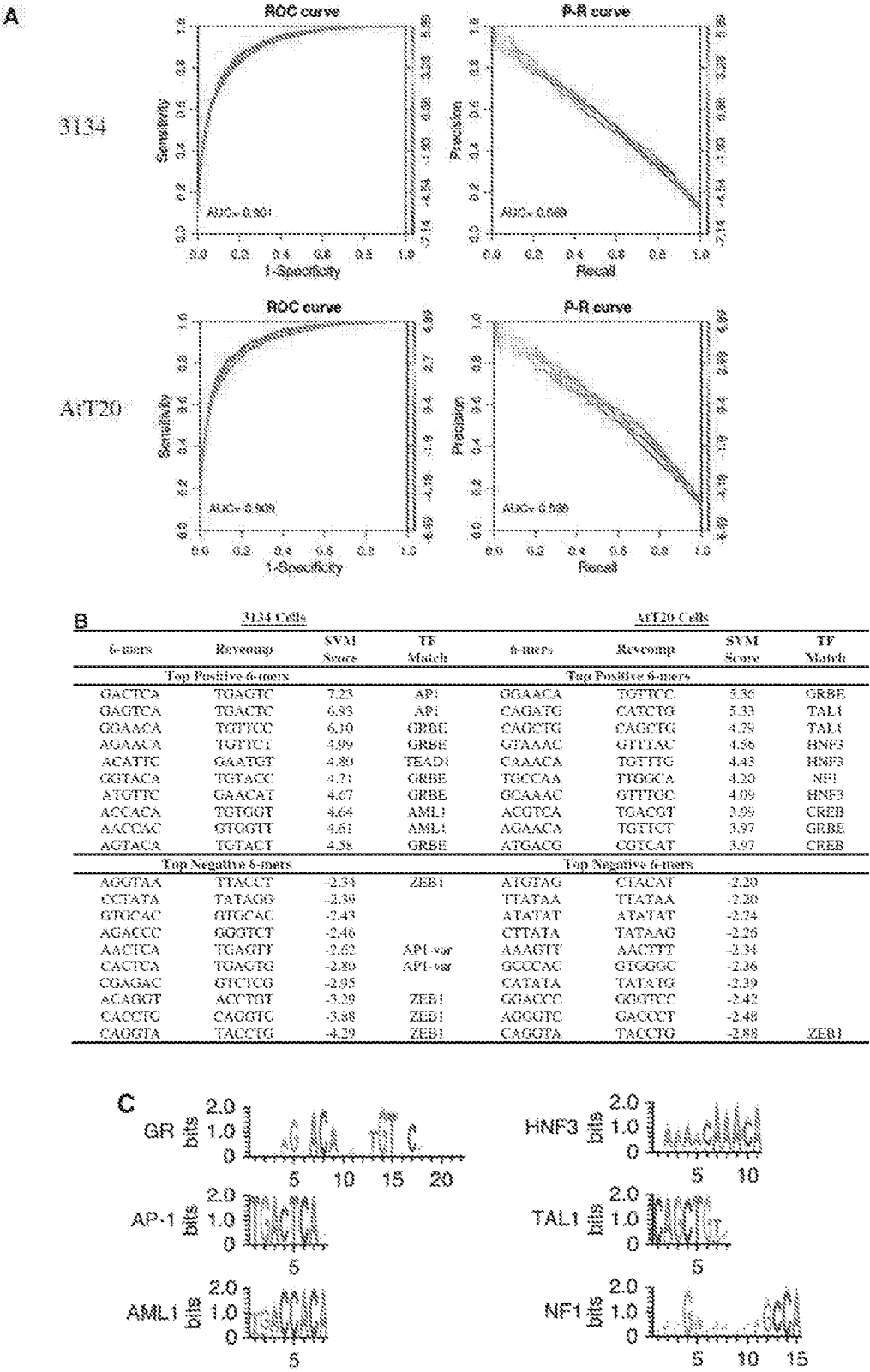
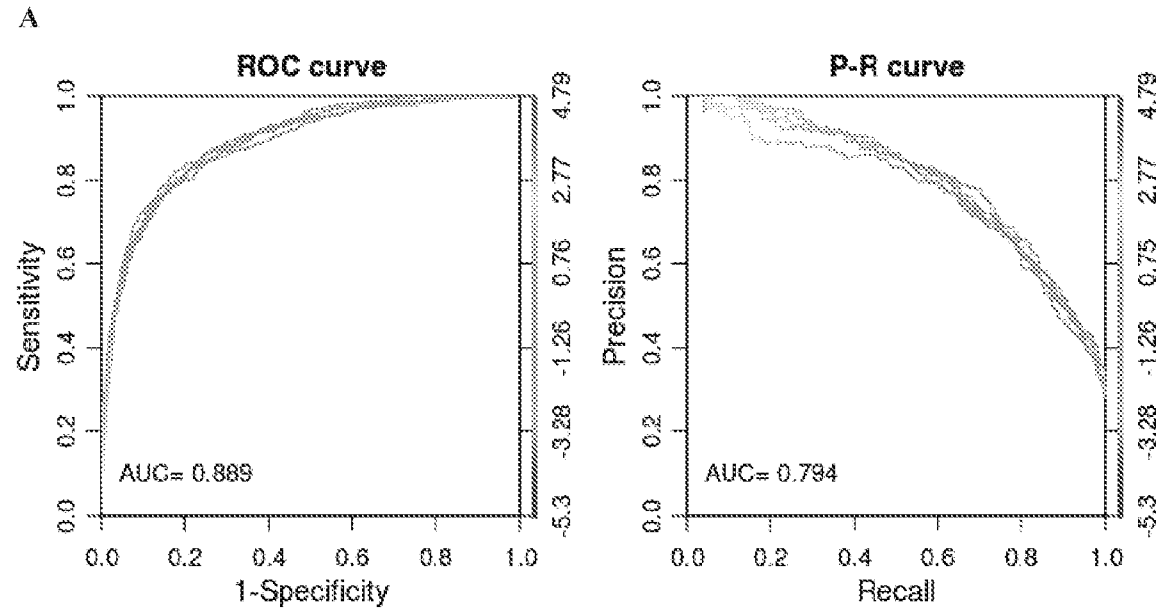


Fig. 34



B

6-mers	Revcomp	SVM Scores	TF Match
Positive 6-mers			
CACCTG	CAGGTG	5.95	ZEB1
CAGATG	CATCTG	4.24	TAL1
CAAACA	TGTTTG	4.15	HNF3
GTAAAC	GTTTAC	4.13	HNF3
ATTTAC	GTAAAT	3.39	HNF3
CAGCTG	CAGCTG	3.36	TAL1
Negative 6-mers			
CCACAA	TTGTGG	-2.62	AML1
AACCAC	GTGGTT	-2.89	AML1
AGTCAT	ATGACT	-3.05	AP1
GACTCA	TGAGTC	-3.75	AP1
GAGTCA	TGACTC	-3.77	AP1
ACCACA	TGTGGT	-4.39	AML1

C

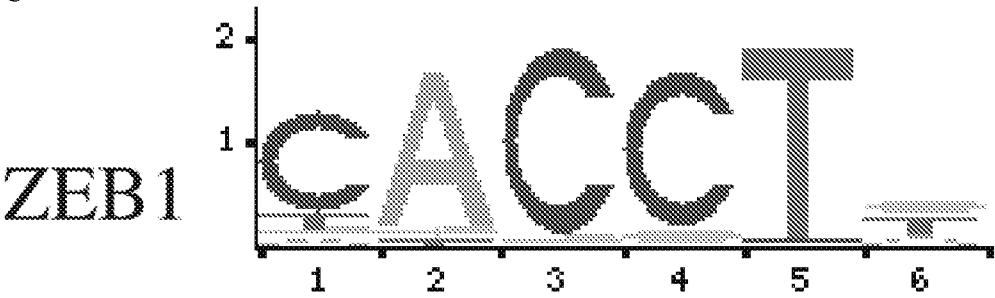


Fig. 35

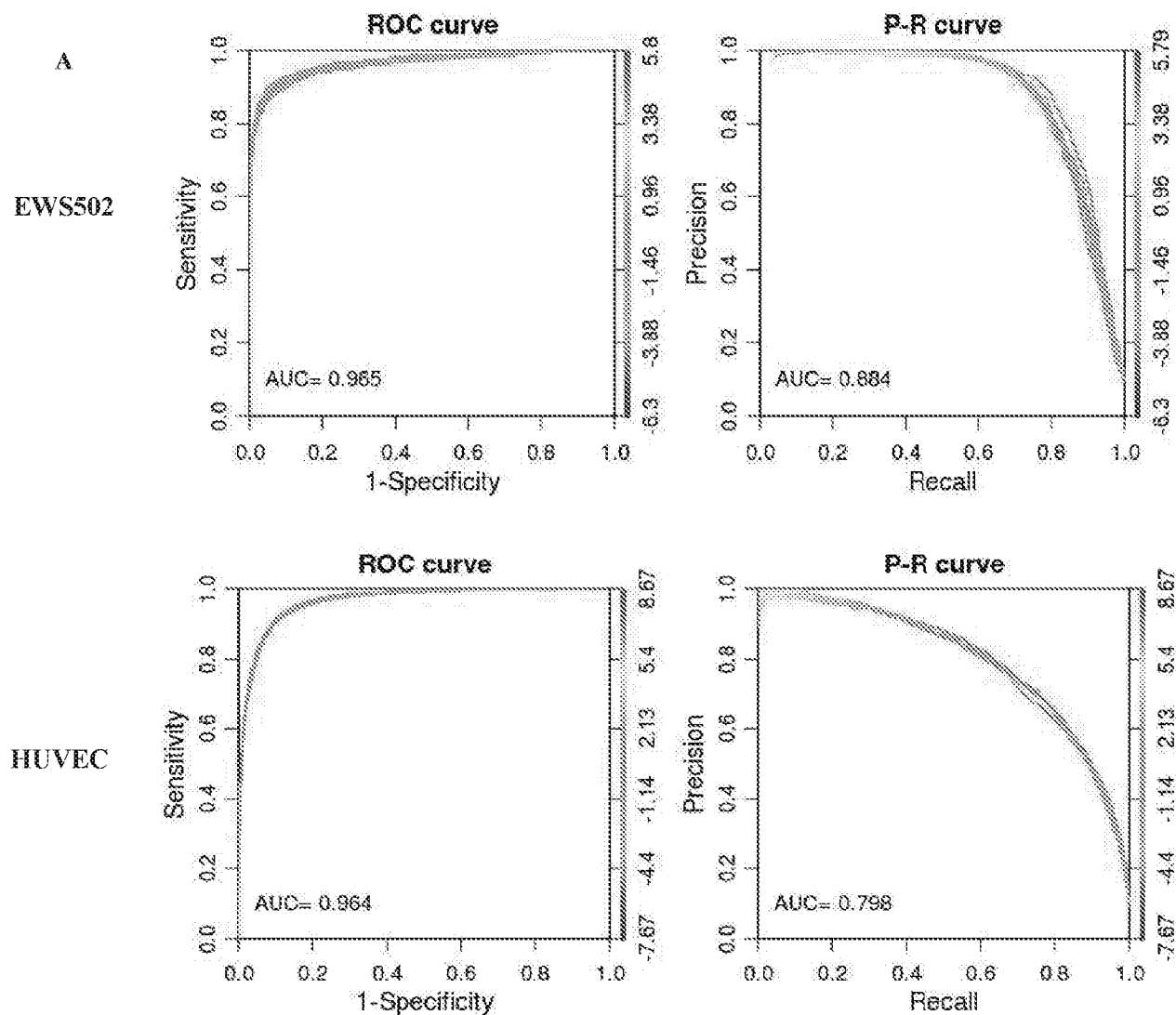


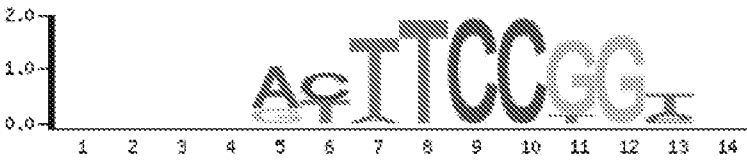
Fig. 36A

B

EWS502				HUVEC			
6-mers	Revcomp	SVM Score	TF Match	6-mers	Revcomp	SVM Score	TF Match
Top Positive 6-mers				Top Positive 6-mers			
CCGGAA	TTCCGG	8.97	Ets	CCGGAA	TTCCGG	9.83	Ets
CAGGAA	TTCCTG	7.49	Ets	CAGGAA	TTCCTG	9.70	Ets
ATTTC	GGAAAT	6.93	Ets	ATTTC	GGAAAT	9.58	Ets
CGGAAA	TTTCCG	6.56	Ets	ACTTCC	GGAAGT	9.45	Ets
AGGAAG	CTTCCT	6.23	Ets	CGGAAA	TTTCCG	8.92	Ets
CGGAAG	CTTCCG	5.72	Ets	AGGAAG	CTTCCT	8.82	Ets
ACTTCC	GGAAGT	5.18	Ets	GACTCA	TGAGTC	7.53	API
ACAGGA	TCCTGT	5.17		CGGAAG	CTTCCG	7.09	Ets
ACCGGA	TCCGGT	4.26	Ets	AGGAAA	TTTCCT	6.70	
AGGAAA	TTTCCT	4.14		ACAGGA	TCCTGT	6.56	
Top Negative 6-mers				Top Negative 6-mers			
CAAGGA	TCCTTG	-1.84	TEAD1	ACGTAT	ATACGT	-2.26	
AGTCCA	TGGACT	-1.92		ACCTGC	GCAGGT	-2.27	ZEB1
AGGTGA	TCACCT	-1.97	ZEB1	AGACTC	GAGTCT	-2.28	
CTGGAC	GTCAG	-1.99		AGGTGA	TCACCT	-2.34	ZEB1
TGGAAG	TTTCCA	-2.29	Ets-var	CTTCCA	TGGAAG	-2.58	Ets-var
ACAGGT	ACCTGT	-2.33	ZEB1	AGGAAT	ATTCTT	-2.82	
AAGGTG	CACCTT	-2.60		AGGTAA	TTACCT	-2.90	ZEB1
AGGAAT	ATTCTT	-2.61	TEAD1	ACAGGT	ACCTGT	-3.10	ZEB1
CTTCCA	TGGAAG	-2.67	Ets-var	CACCTG	CAGGTG	-3.24	ZEB1
CAGGTA	TACCTG	-3.28	ZEB1	CAGGTA	TACCTG	-4.45	ZEB1

C

ETS (FLI1)



TEAD1

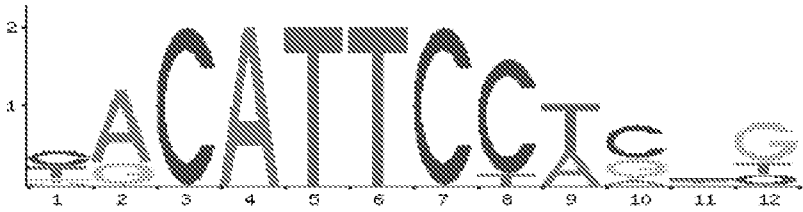


Fig. 36B-36C

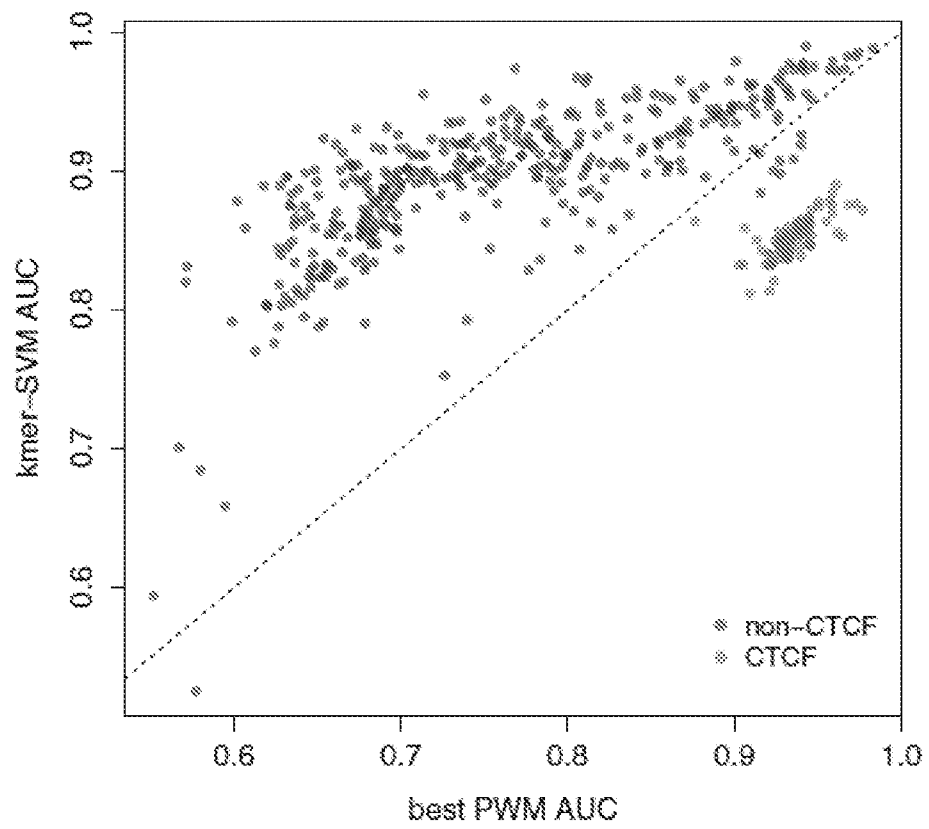


Fig. 37

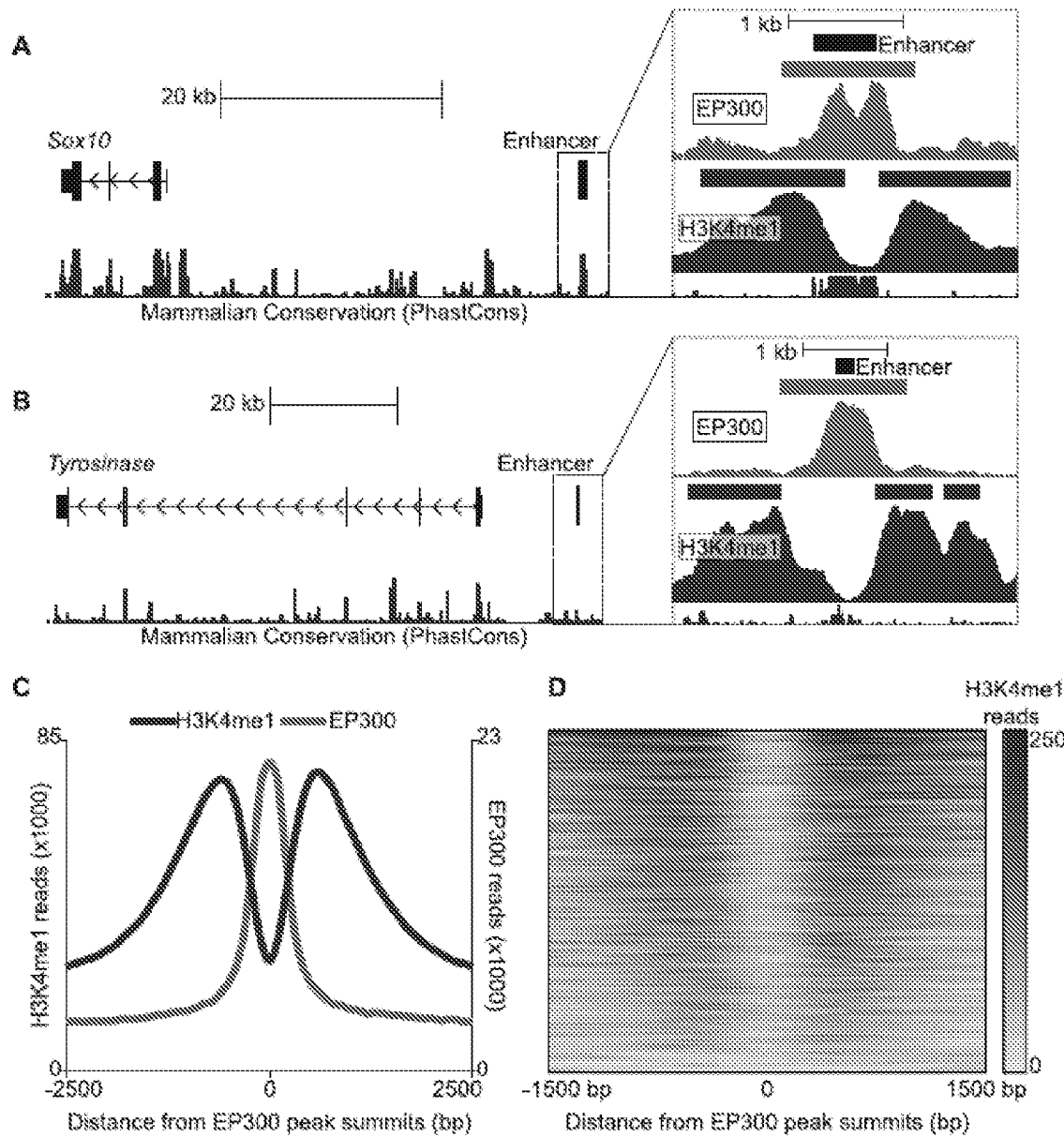


Fig. 38

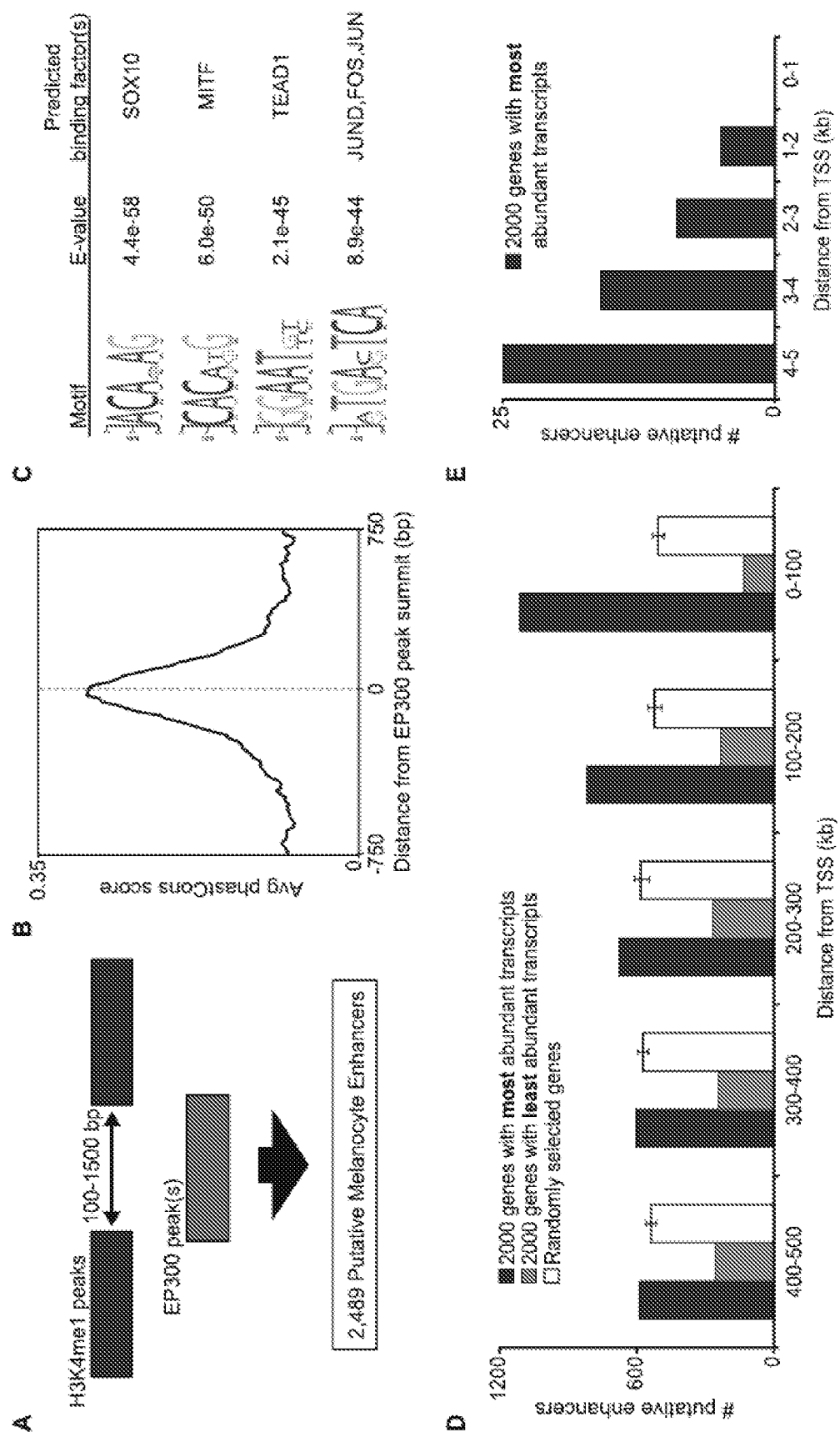


Fig. 39

Table 12: Gene Ontology (GO) terms associated with genes proximal to putative melanocyte enhancers

GO category	Melanocyte-related GO term	P-value
Biological process	Pigmentation	1.3×10^{-7}
	Melanocyte differentiation	1.5×10^{-7}
	Pigment cell differentiation	2.0×10^{-7}
	Developmental pigmentation	3.1×10^{-7}
	Melanin metabolic process	6.5×10^{-5}
Mouse phenotype	Abnormal hair follicle melanocyte morphology	3.1×10^{-9}
	Abnormal melanocyte morphology	1.5×10^{-8}
	Abnormal extracutaneous pigmentation	6.8×10^{-8}
	Hypopigmentation	8.3×10^{-7}
	Abnormal hair follicle melanin granule morphology	1.8×10^{-5}
	Absent hair follicle melanin granules	2.3×10^{-5}
	White spotting	2.7×10^{-5}
	Abnormal melanosome morphology	7.4×10^{-5}
	Abnormal melanoblast migration	9.3×10^{-5}
	Yellow coat color	6.4×10^{-4}
Disease ontology	Melanoma	4.1×10^{-19}
	Melanocytic neoplasm	7.4×10^{-19}
	Waardenburg syndrome	9.3×10^{-6}
Cellular component	Melanosome	1.3×10^{-11}

Fig. 40

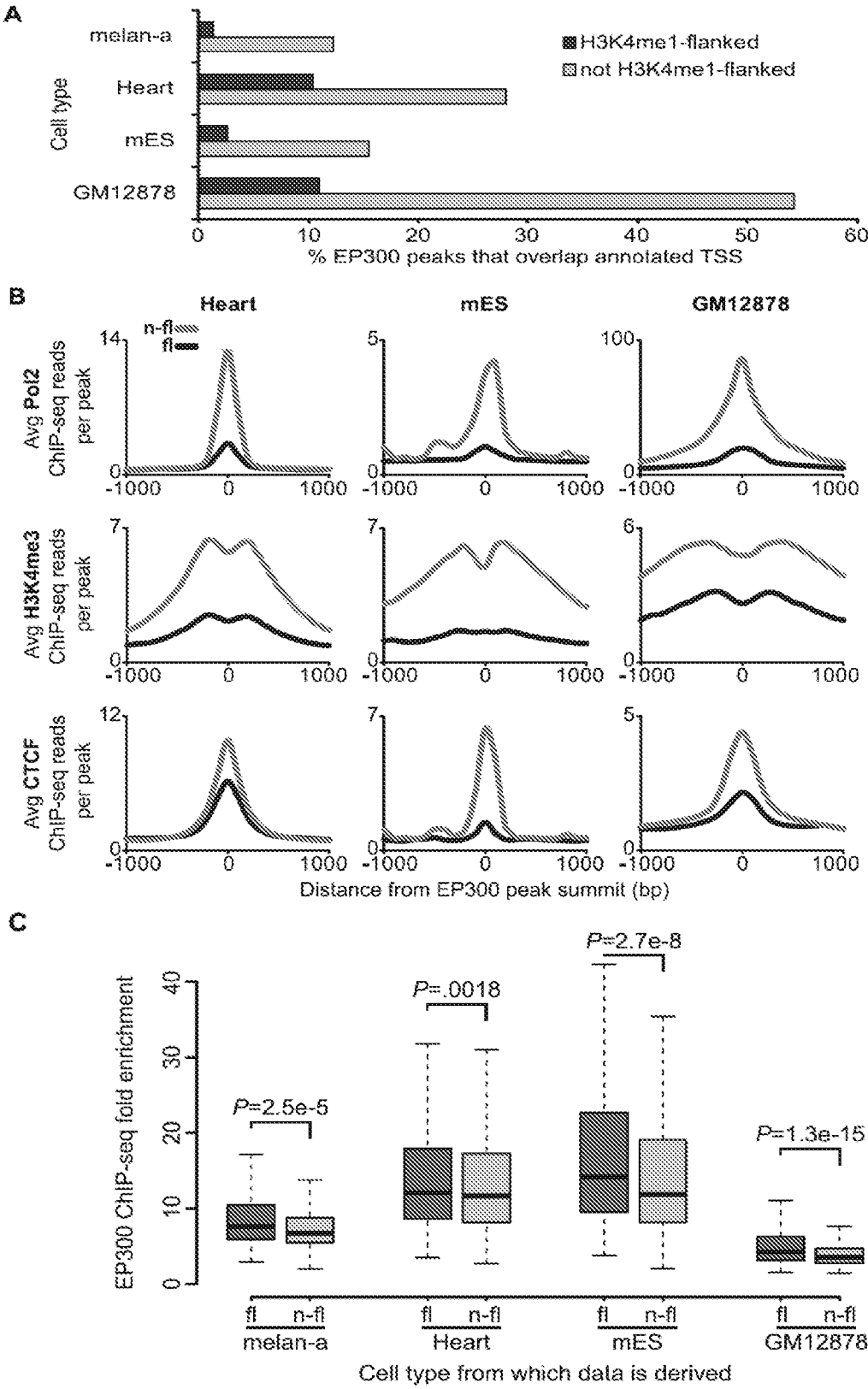


Fig. 41

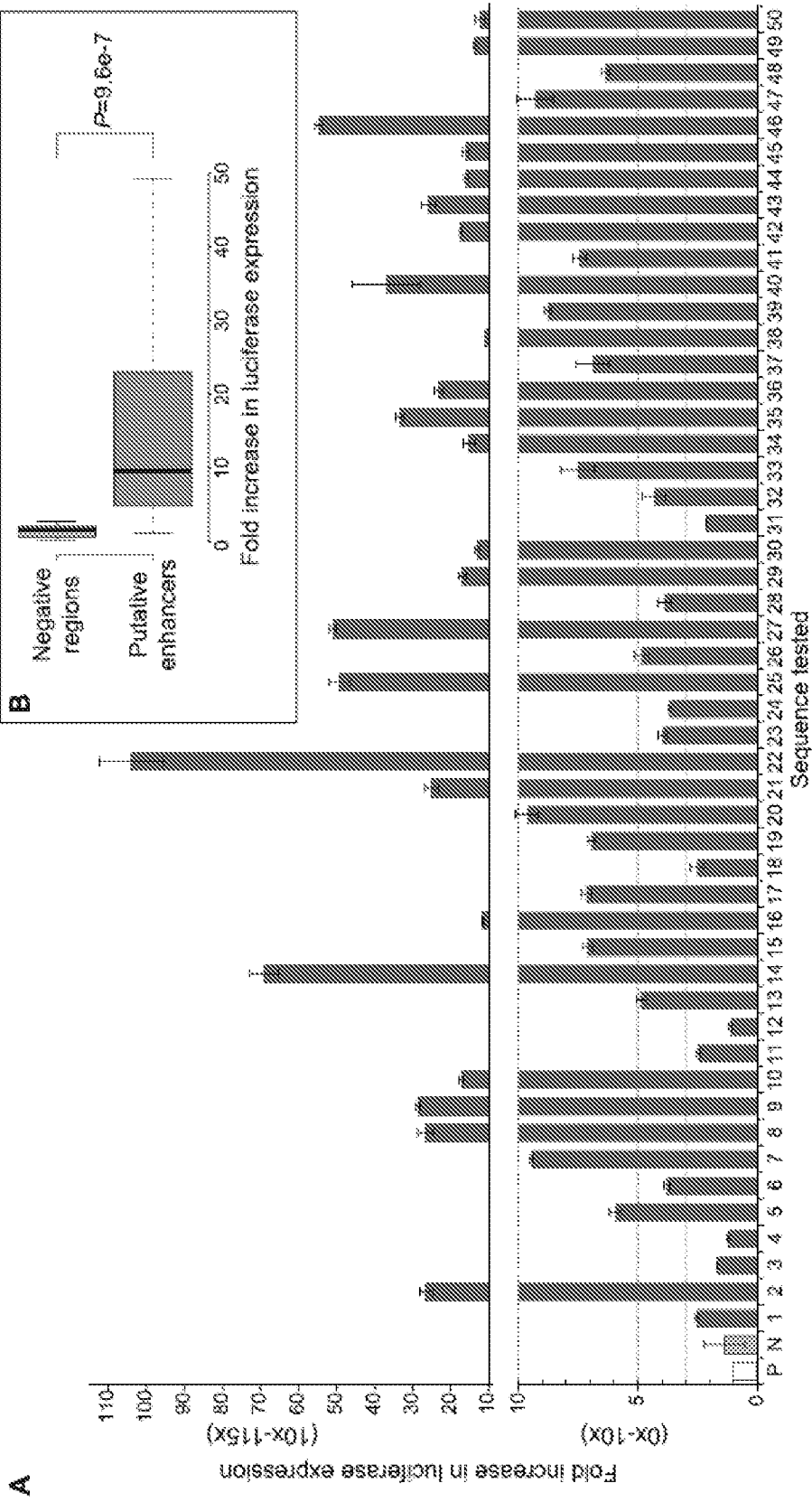
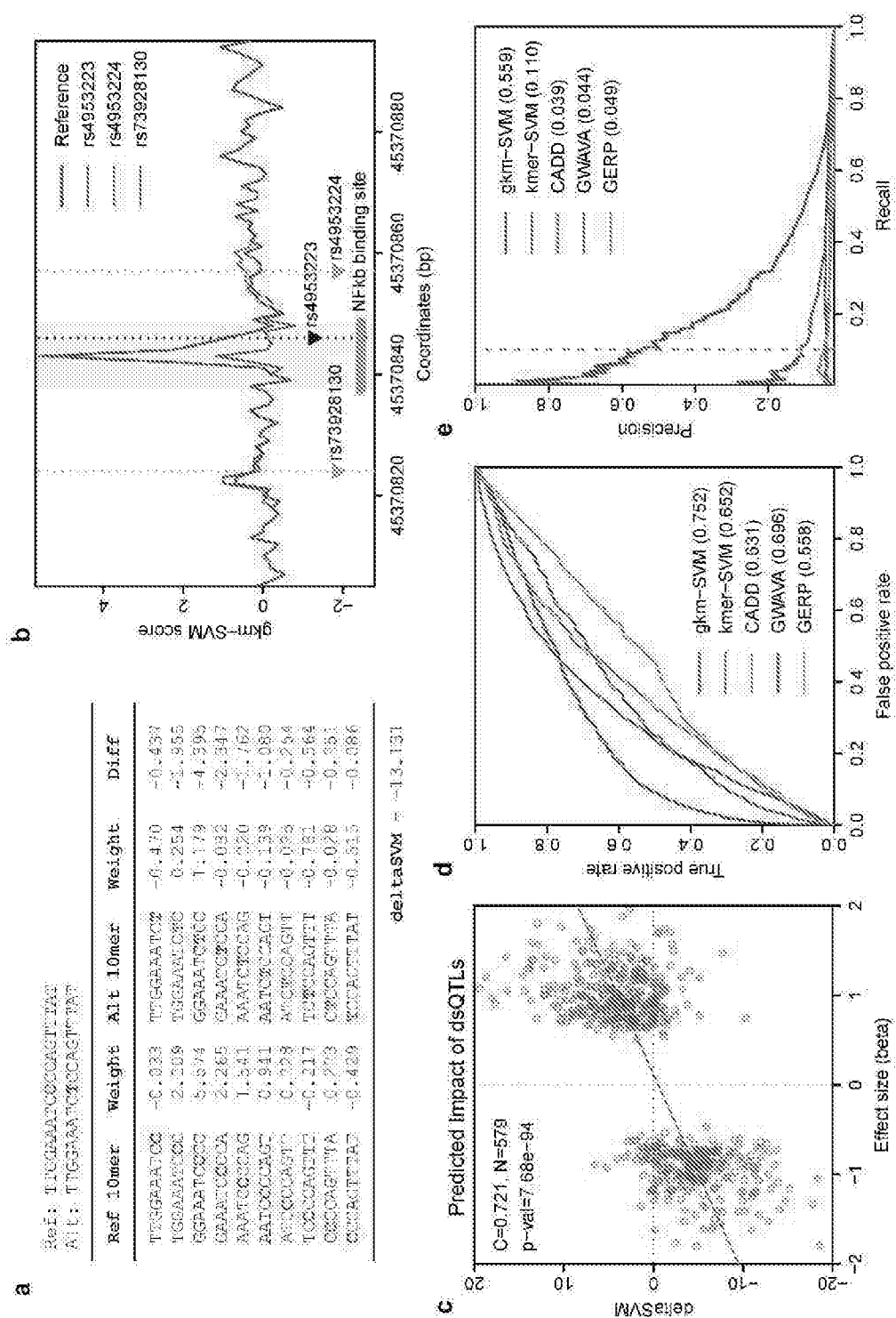


Fig. 42



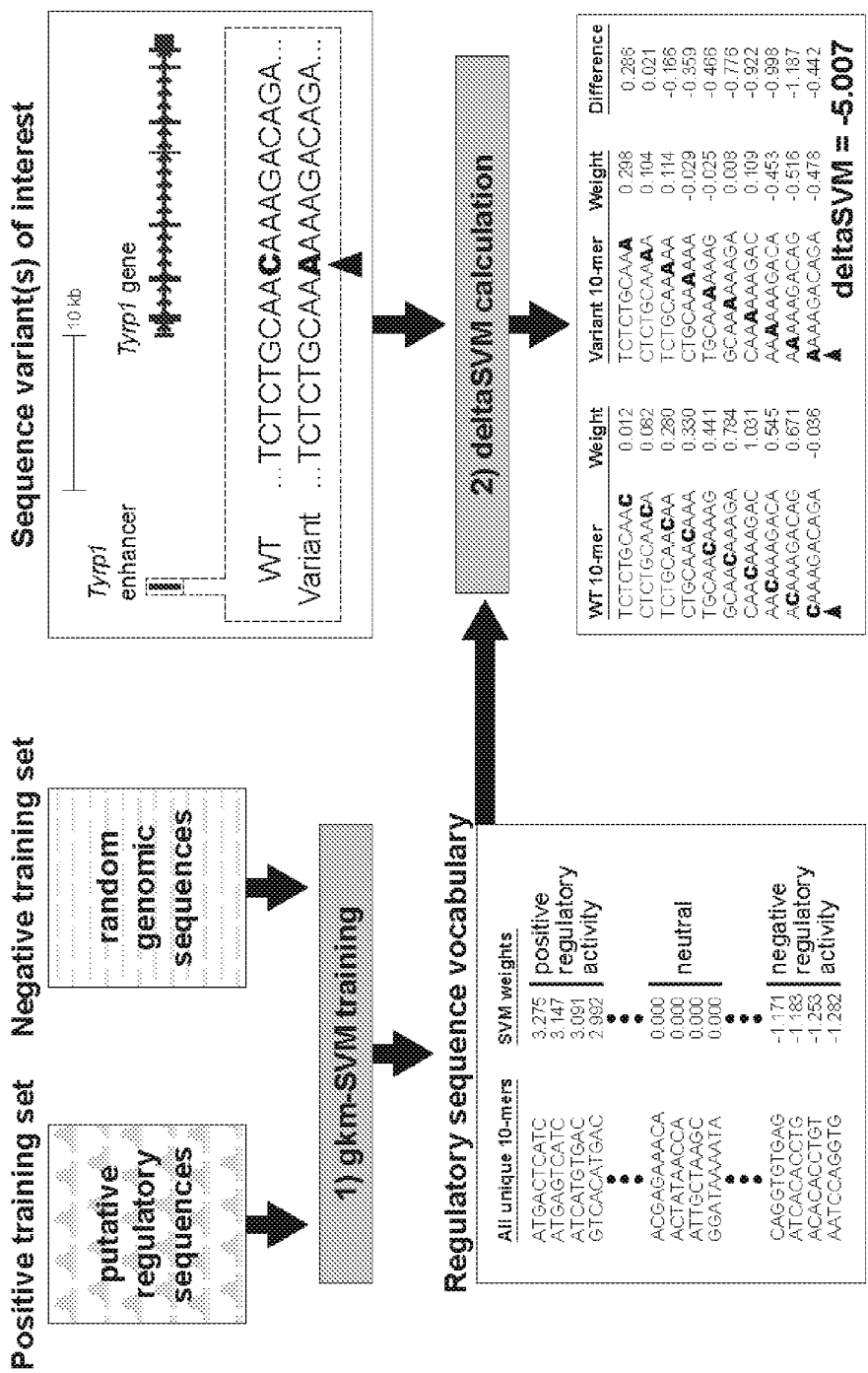


Fig. 44

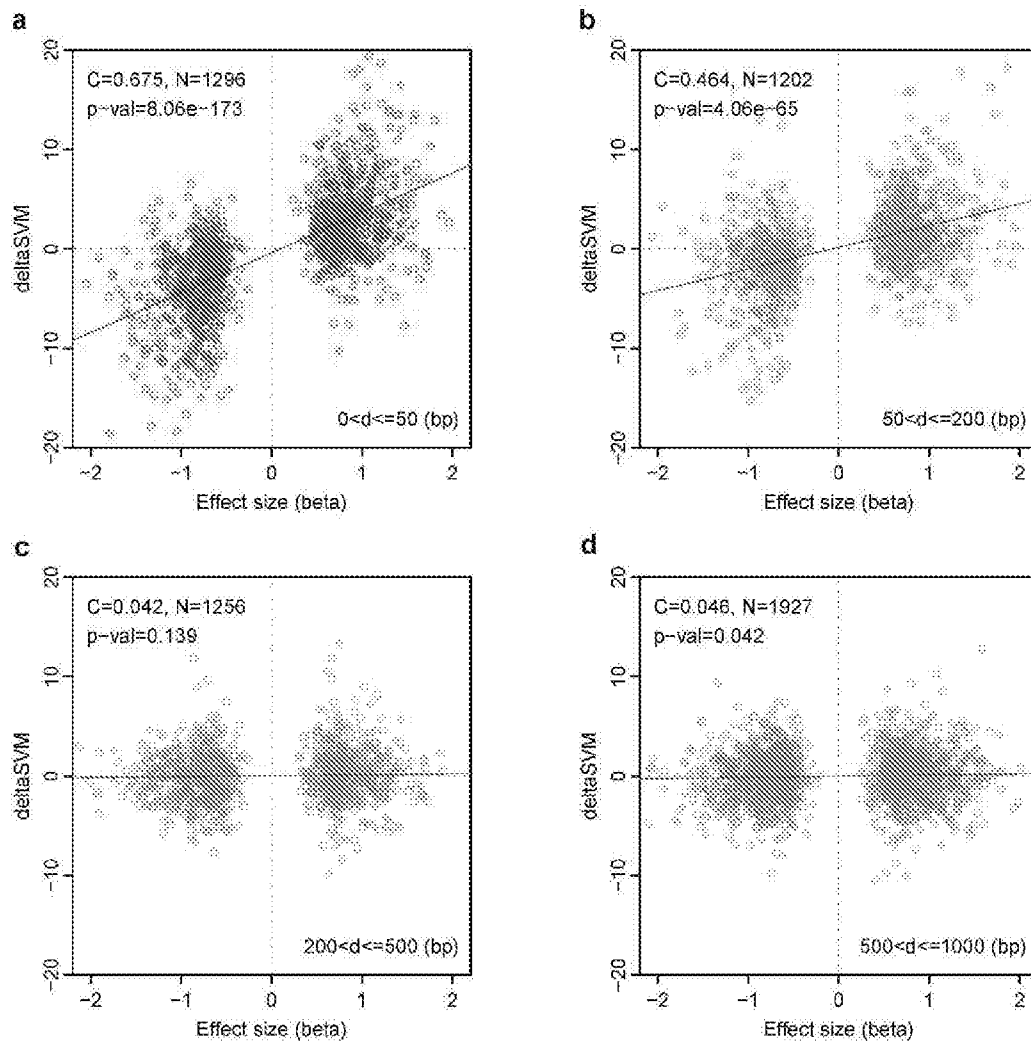


Fig. 45

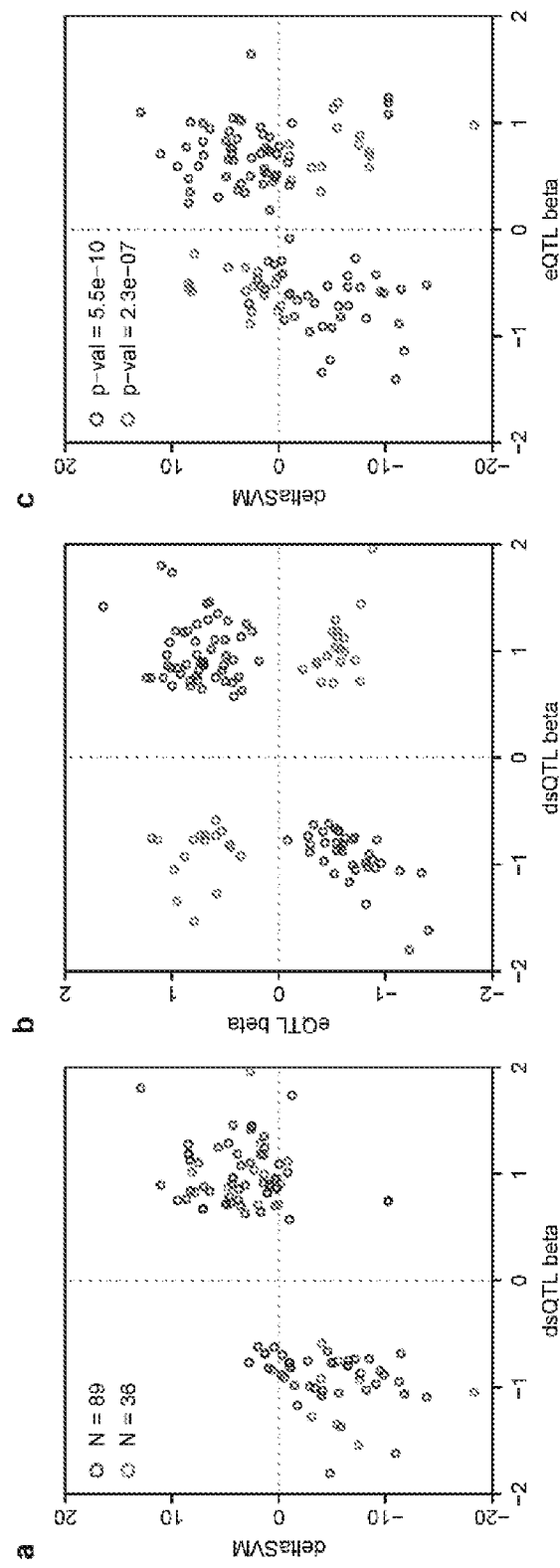


Fig. 46

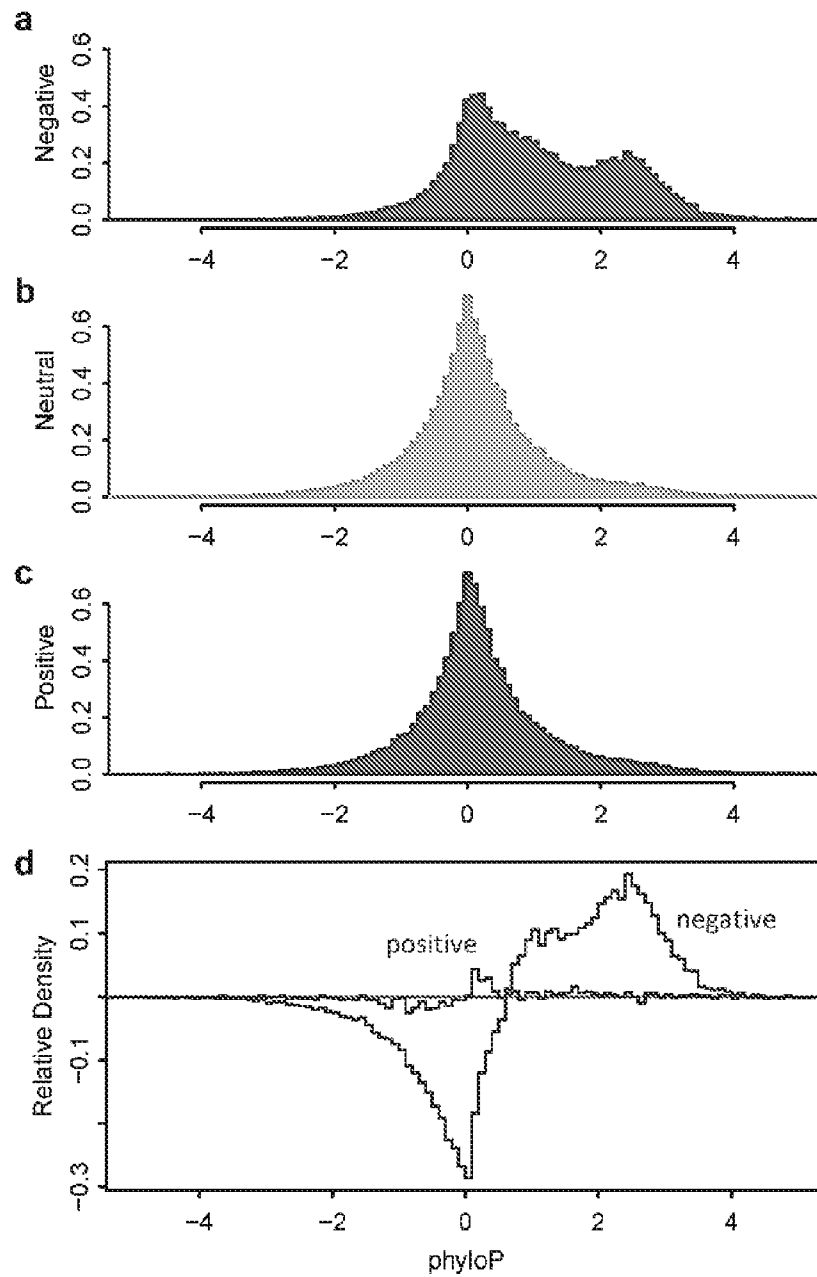


Fig. 47

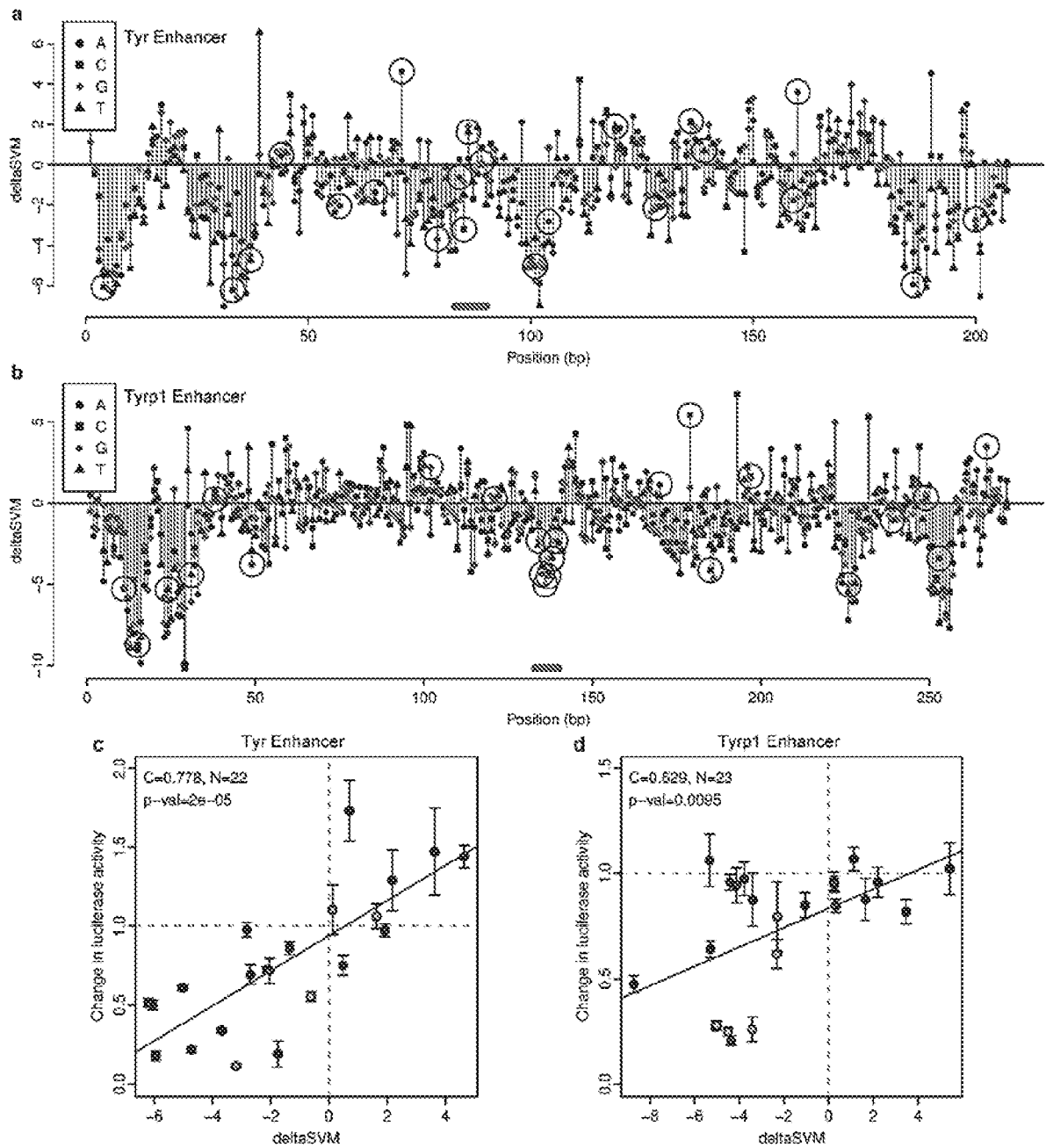


Fig. 48

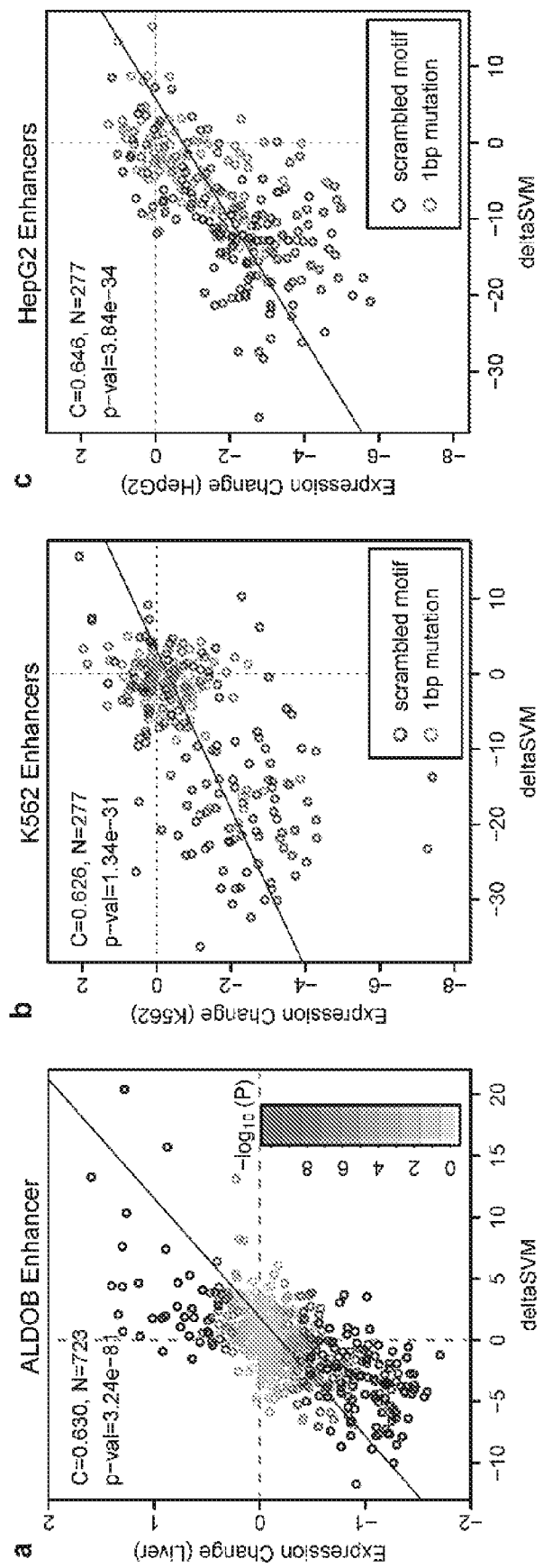


Fig. 49

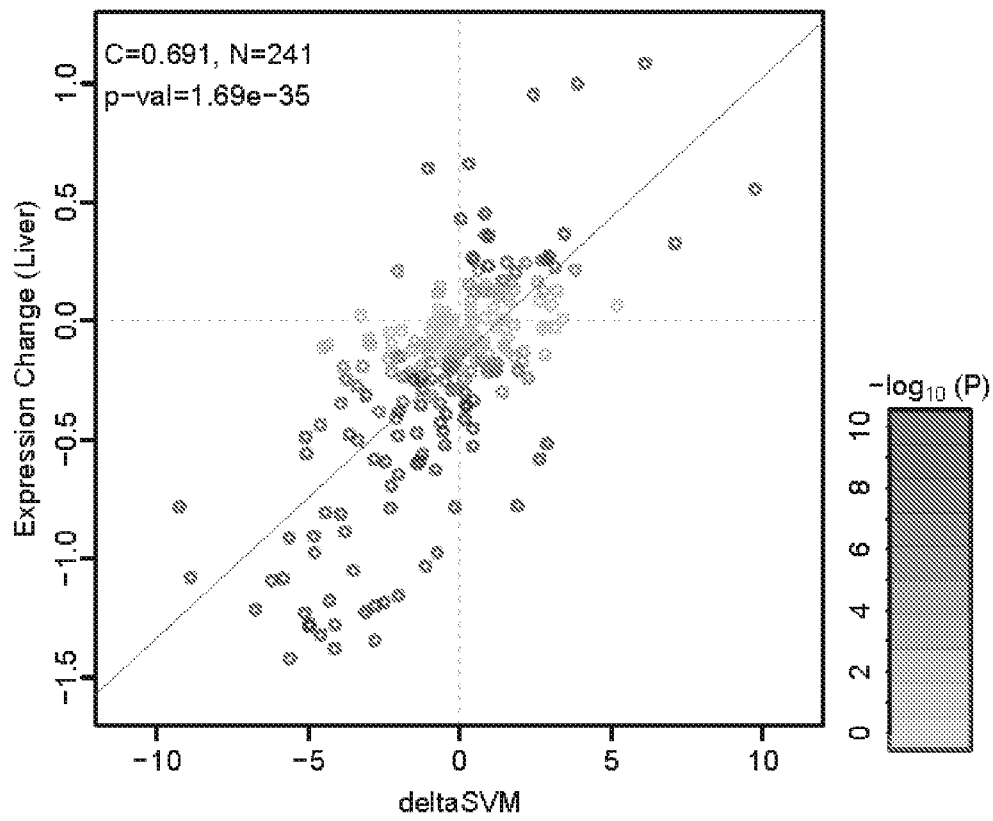


Fig. 50

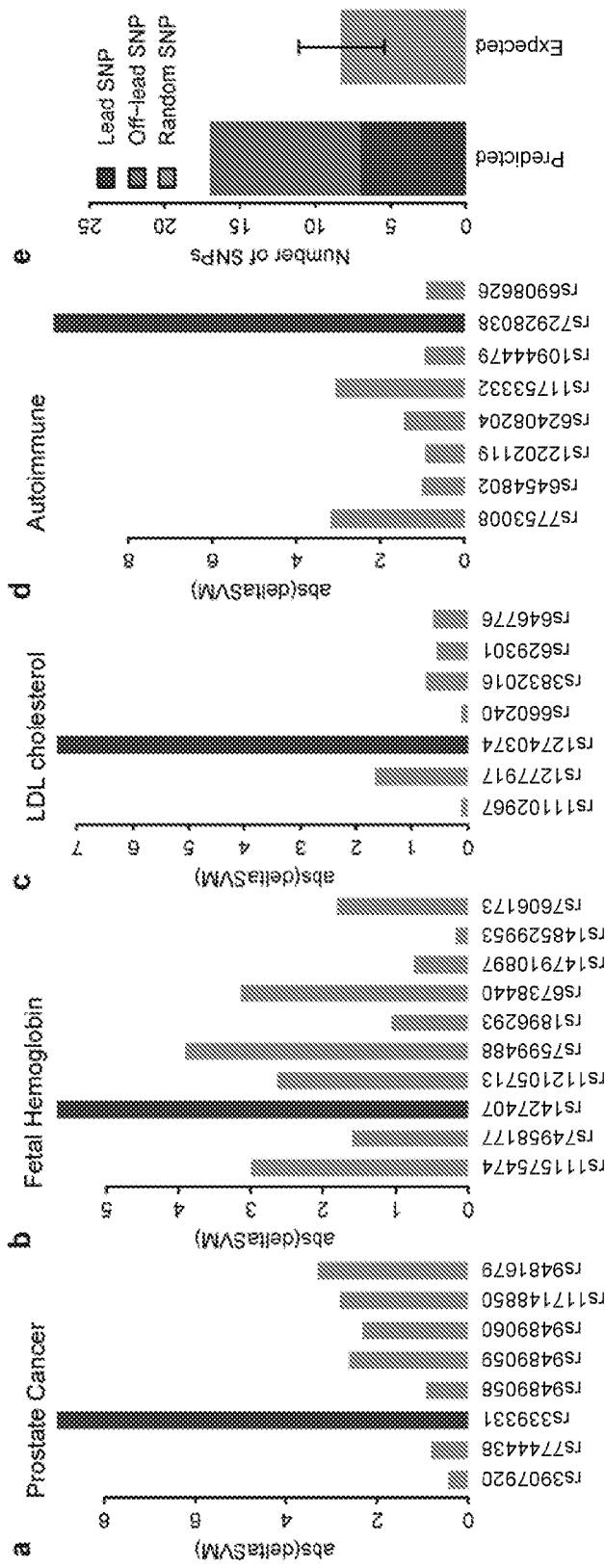


Fig. 51 A-E

deltaSVM predicted causal autoimmune SNPs

Disease	Lead SNP	Predicted SNP	CHR	Lead SNP Position	Predicted SNP Position	Distance	Nearest Gene	deltaSVM	Ref.
VIT	rs853308	rs860475	chr8	133929917	133929799	-118	TG	16.797	29
VIT	rs16872571	rs4697651	chr4	10726853	10721433	-5420	.	13.978	29
T1D	rs7020673	rs4380994	chr9	4291747	4282536	-9211	GLIS3	13.591	30
CRO	rs6908425	rs7748720	chr6	20728731	20689945	-38786	CDKAL1	13.177	31, 32
MS	rs1782645	rs1250568	chr10	81048611	81045280	-3331	ZMIZ1	11.039	33
CEL	rs1250552	rs1250568	chr10	81058027	81045280	-12747	ZMIZ1	11.039	34
CRO	rs10801047	rs6665749	chr1	191559356	191577594	18238	.	10.646	35
CRO	rs2797685	rs2797685	chr1	7879063	7879063	0	PER3	10.411	31
Allergy	rs962993-T	rs962993	chr10	9053132	9053132	0	.	10.008	36
PBC	rs10931468	rs3771317	chr2	191538562	191543962	5400	NAB1	9.999	37
CEL	rs7753008	rs72928038	chr6	90809639	90976768	167129	BACH2	9.787	38
RA									
ATD	rs72928038	rs72928038	chr6	90976768	90976768	0	BACH2	9.787	39, 40, 33
MS									
MS	rs564976	rs485789	chr3	159729059	159730148	1089	AK097161	8.782	
MS	rs733724	rs733724	chr6	105223864	105223864	0	HACE1	7.360	41
PBC	rs3024921	rs3024921	chr2	191943272	191943272	0	STAT4	6.931	42

Fig. 51F

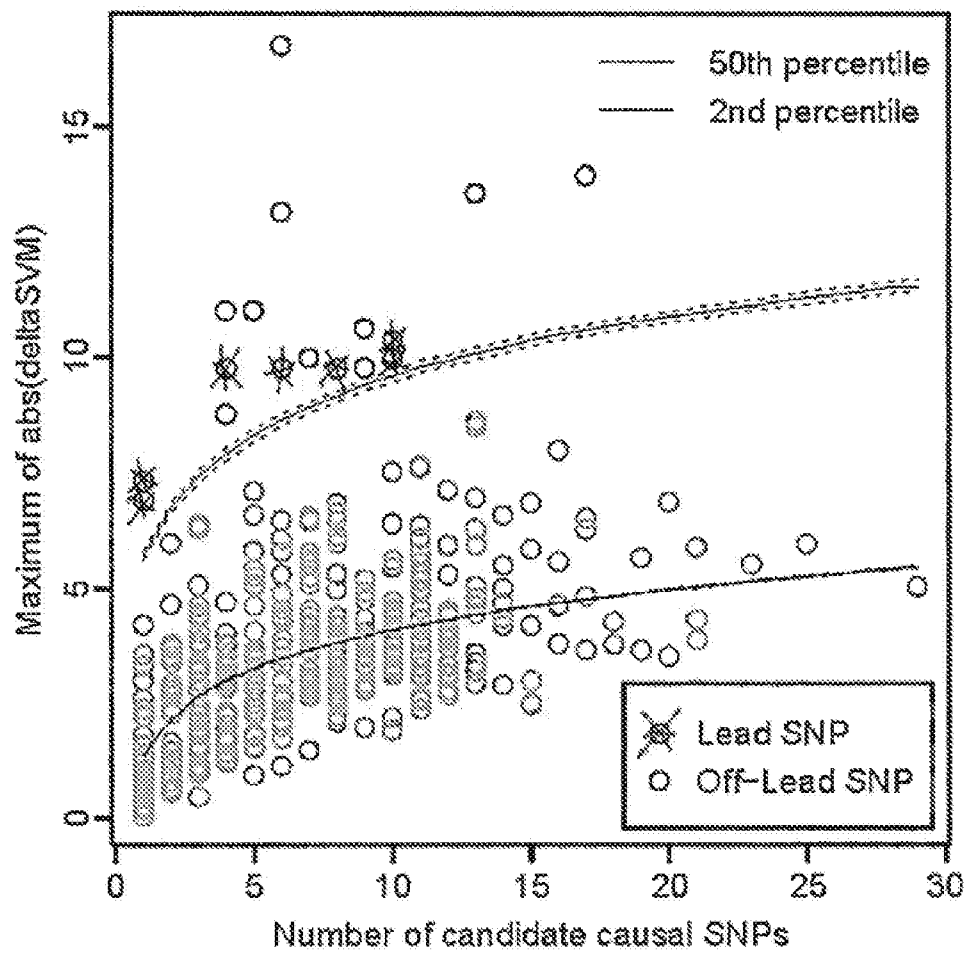


Fig. 52

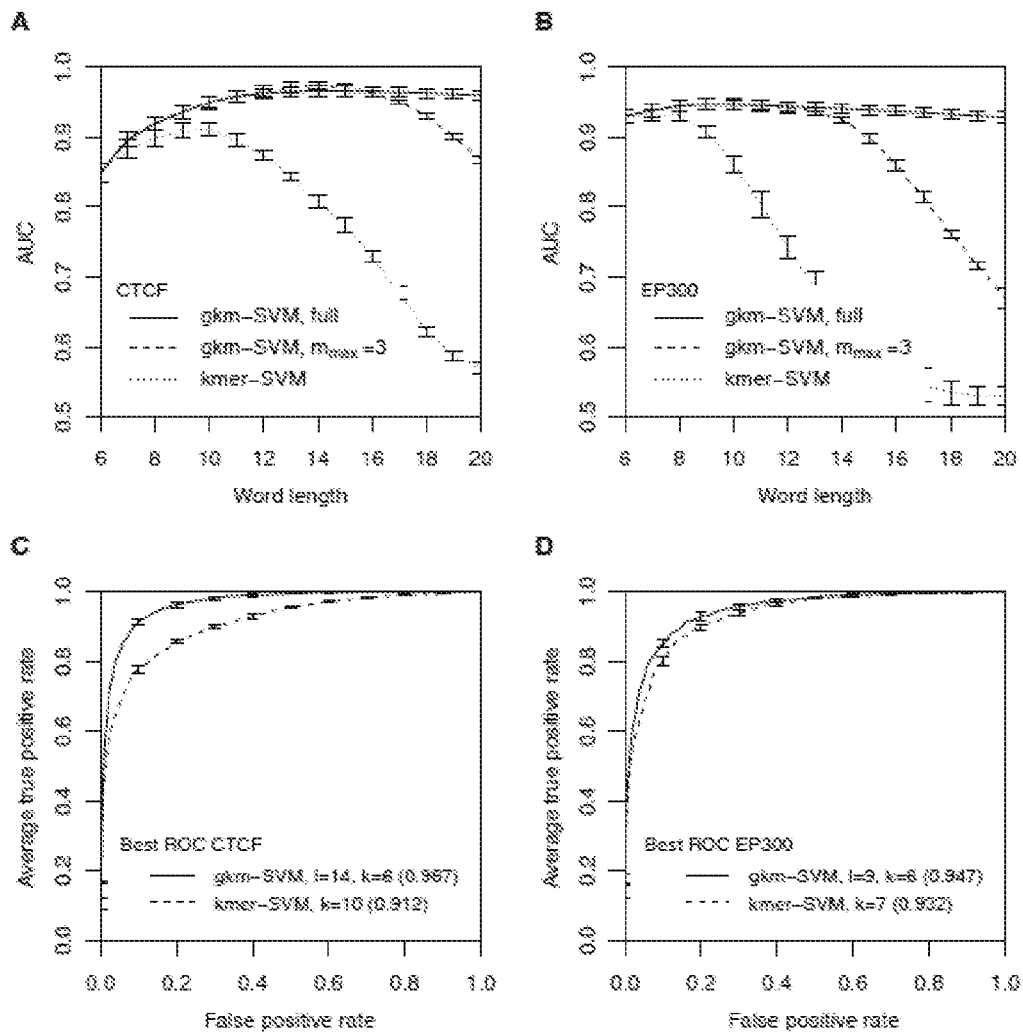


Fig. 53A-D

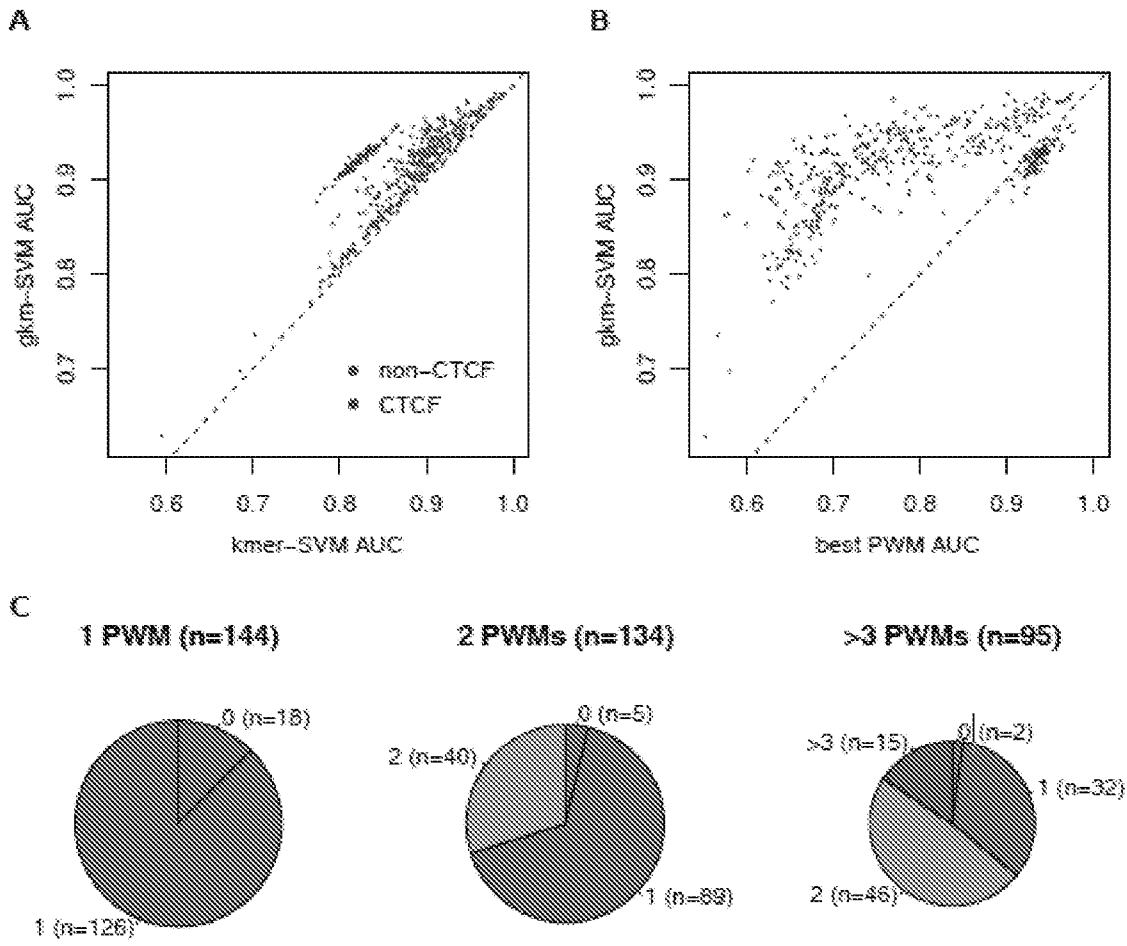


Fig. 54A-C

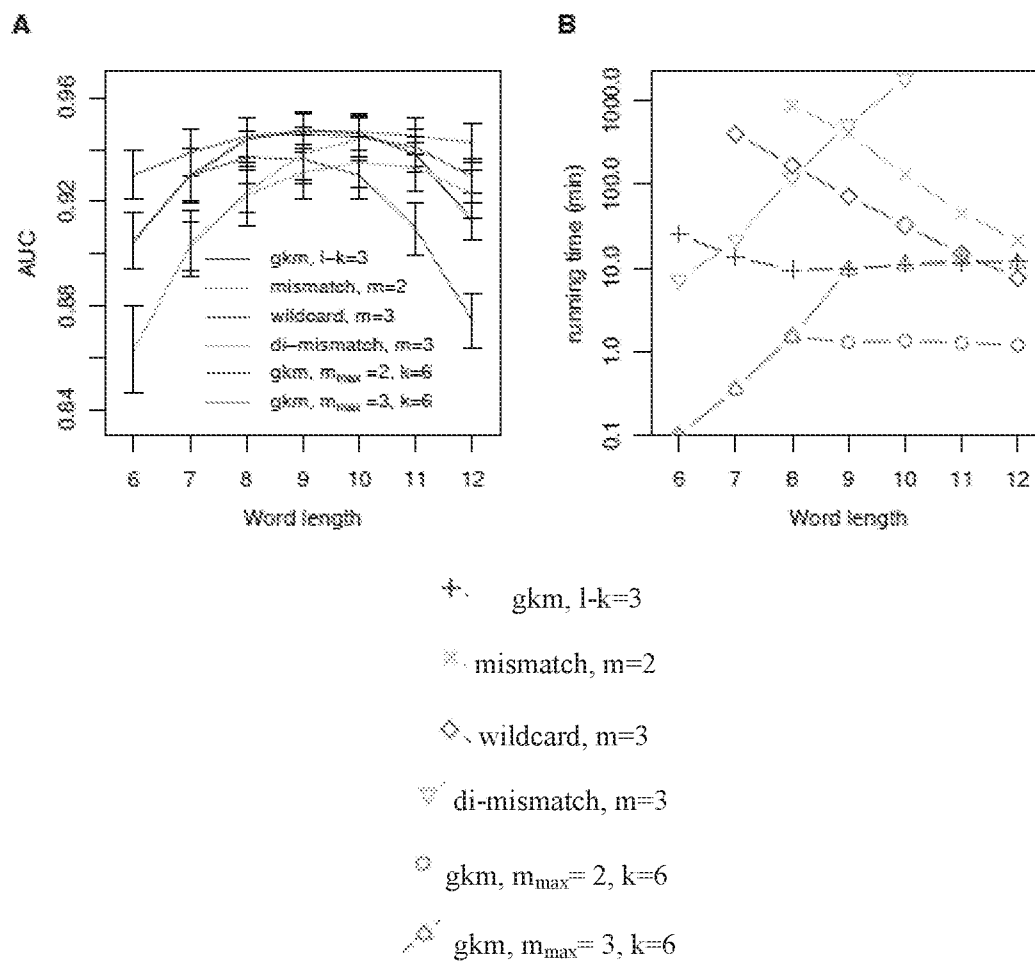


Fig. 55 A-B

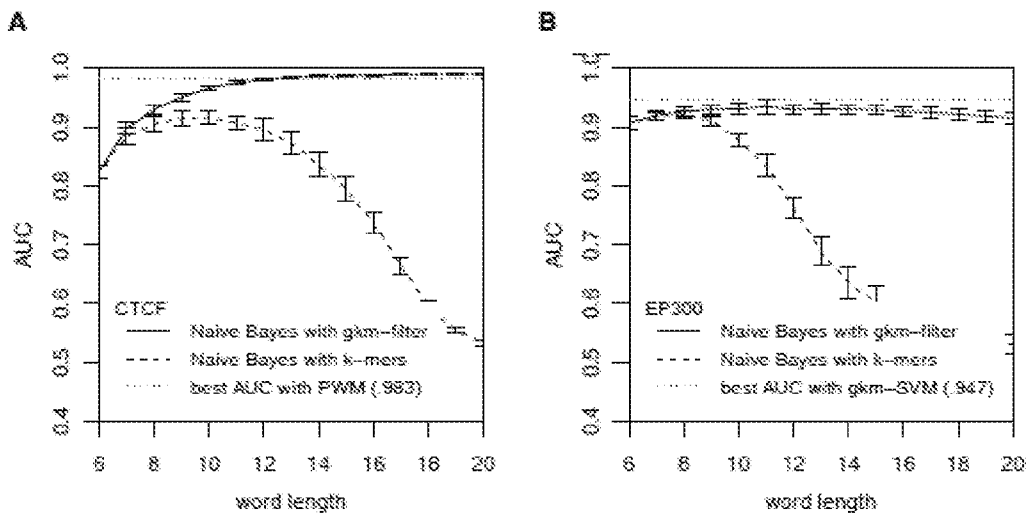


Fig. 56 A-B

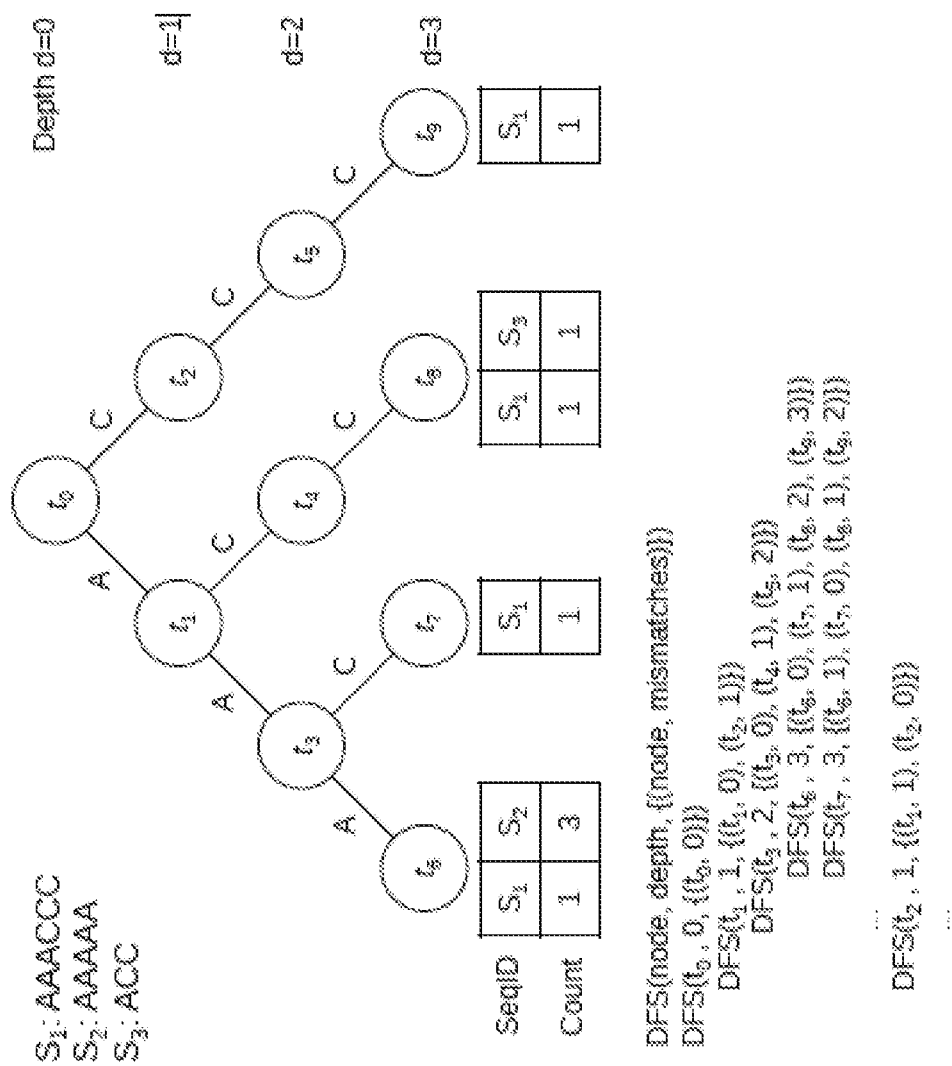


Fig. 57

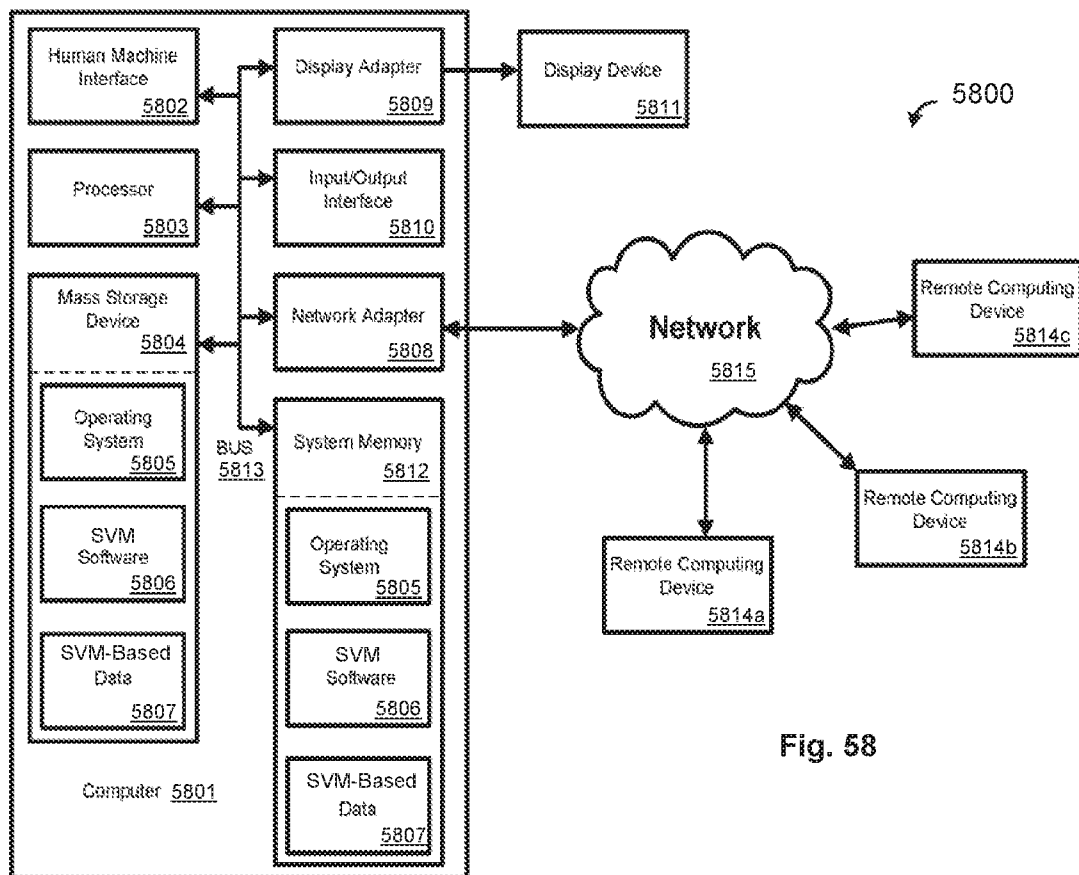


Fig. 58

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US2016/032163

A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - G01N 33/53; G06F 19/18; G06F 19/24 (2016.01)

CPC - G01N 33/53; G06F 19/18; G06F 19/24 (2016.08)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC - G01N 33/53; G06F 19/18; G06F 19/24

CPC - G01N 33/53; G06F 19/18; G06F 19/24

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

USPC - 435/7.1; 506/9; 702/19 (keyword delimited)

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

Patbase, Google Patents, PubMed, Google

Search terms used: SVM vector allele weight computer screen assay autoimmune SNP enhancer k-mer

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2014/0129152 A1 (BEER et al) 08 May 2014 (08.05.2014) entire document	1-15
Y	US 2012/0140636 A1 (RESENDE et al) 07 June 2012 (07.06.2012) entire document	1-15
Y	US 2012/0309639 A1 (HAKONARSON et al) 06 December 2012 (06.12.2012) entire document	12, 13
A	US 2013/0073213 A1 (CENTOLA et al) 21 March 2013 (21.03.2013) entire document	1-15
P,X	LEE et al. "A method to predict the impact of regulatory variants from DNA sequence," Nat. Genet. 15 June 2015 (15.06.2015), Vol. 47, No. 8, Pgs. 955-961. entire document	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

01 August 2016

Date of mailing of the international search report

22 AUG 2016

Name and mailing address of the ISA/

Mail Stop PCT, Attn: ISA/US, Commissioner for Patents

P.O. Box 1450, Alexandria, VA 22313-1450

Facsimile No. 571-273-8300

Authorized officer

Blaine R. Copenheaver

PCT Helpdesk: 571-272-4300

PCT OSP: 571-272-7774