SArKS: Discovering gene expression regulatory motifs and domains by suffix array kernel smoothing

Dennis Wylie[1*], Hans A. Hofmann[1,2,3,4], Boris V. Zemelman[2,4,5,6**]

1 Center for Computational Biology and Bioinformatics, University of Texas at Austin
2 Institute for Cellular and Molecular Biology, University of Texas at Austin
3 Department of Integrative Biology, University of Texas at Austin
4 Institute for Neuroscience, University of Texas at Austin
5 Department of Neuroscience, University of Texas at Austin
6 Center for Learning and Memory, University of Texas at Austin

* denniswylie@austin.utexas.edu
** zemelmanb@mail.clm.utexas.edu

Abstract

We describe an algorithm that can mine differential gene expression data to identify candidate cell type-specific DNA regulatory sequences. Differential expression is usually quantified as a continuous score—fold-change, test-statistic, p-value—comparing biological classes. Unlike existing approaches, our de novo strategy, termed SArKS, applies nonparametric kernel smoothing to uncover promoter motifs that correlate with elevated differential expression scores. SArKS detects motifs by smoothing sequence scores over sequence similarity. A second round of smoothing over spatial proximity reveals multimotif domains (MMDs). Discovered motifs can then be merged or extended based on adjacency within MMDs. False positive rates are estimated and controlled by permutation testing. We apply SArKS to a published gene expression data set representing distinct neocortical neuron classes in *M. musculus*. When benchmarked against several existing algorithms for correlative motif discovery using a cross-validation procedure, SArKS identified larger motif sets that formed the basis for regression models with higher correlative power. Finally, we examined specific motifs derived by SArKS from putative promoters of genes enriched in parvalbumin-expressing GABAergic interneurons, identifying several known biologically meaningful sequence elements.

1 Introduction

Discrete sequences—of tones, of symbols, or of molecular building blocks—can provide clues to other characteristics of the entities from which they are derived: a phrase in a bird's song can reveal which species it belongs to, the use of an idiomatic expression can pinpoint a speaker's geographic origin, and a specific short string of nucleotide residues can illuminate the function of a DNA domain. In these examples, insights are gleaned from informative *motifs*—short subsequences that match some frequently recurring discernible pattern.

Of particular interest are DNA regions modulating differential gene expression. Identification of motifs within such regions can help to uncover DNA sequences with autonomous and predictable function (stereotyped expression regulation). While prior efforts have focused on short motifs that participate in regulating eukaryotic gene expression [Walhout, 2006], the details of how and which motifs are needed for expression specificity remain poorly understood.

We present a broadly-applicable algorithm for identifying DNA regulatory regions that support differential gene expression. Our strategy is predicated on the following suppositions: (a) gene expression regulatory regimes involve the binding of transcription factors (TFs) to sites on non-coding DNA in the vicinity of a transcription start site (TSS) [Maston *et al.*, 2006, Nguyen and D'haeseleer, 2006]; (b) TFs act combinatorially to attract and repel transcription machinery [Walhout, 2006]; (c) the same TF binding site may appear multiple times within a stretch of DNA, interspersed with other binding sites [Gotea *et al.*, 2010]; and (d) there is more than one solution: different genes, even those co-expressed within a single cell, may rely on different regulatory mechanisms [Badis *et al.*, 2009]. In accord with these suppositions, we aim to identify TF binding sites associated with enriched transcripts and scrutinize their arrangement for significant patterns that can then be evaluated experimentally.

Many different motif identification methods have been described. Consensus-based methods such as Weeder [Pavesi *et al.*, 2001, Pavesi *et al.*, 2004] focus on fixed-length motifs that occur repeatedly (allowing for small numbers of mismatches) in sequences of interest and can be efficiently implemented using suffix trees [Sagot, 1998, Marsan and Sagot, 2000, Pavesi *et al.*, 2001]. Alternately, profile-based methods such as MEME [Bailey and Elkan, 1995, Bailey *et al.*, 2006, Bailey *et al.*, 2009] build a probabilistic motif profile to be compared to a background model in order to classify subsequences as either matching the motif or not.

In contrast, discriminative motif analysis [Sinha, 2003] identifies motifs specifically differentiating one set of sequences (e.g., promoter regions for genes with a given expression pattern) from another (e.g., a set of reference promoter regions). A number of approaches have been applied to this problem [Segal *et al.*, 2002, Segal and Sharan, 2005, Redhead and Bailey, 2007, Fauteux *et al.*, 2008, Valen *et al.*, 2009, Huggins *et al.*, 2011, Yao *et al.*, 2014]; one popular recent example, DREME [Bailey, 2011], employs Fisher's exact test to compare counts of motif matches in the target/positive sequences with counts in the background/negative sequences (followed by further refinement of motif profiles for satisfactory candidate motifs). An alternative discriminative strategy employing similar hypergeometric enrichment calculations but coupling them with a zero-or-one-occurrence-per-sequence (ZOOPS) scoring approach is implemented in HOMER [Heinz *et al.*, 2010].

Discriminative approaches incorporate gene-specific information beyond promoter sequence into the motif discovery process—e.g., whether or not a gene shows elevated expression following an experimental manipulation—but these methods implicitly assume a binary representation (e.g., elevated vs. not elevated) for such covariates. Given that the information used to establish the contrasting gene sets is often obtained in the form of continuous measures of differential expression (t-statistics, f-statistics, etc.), with some genes exhibiting extremely divergent expression patterns across conditions while others show more modest differences, it is also useful to consider "correlative motif discovery," which seeks motifs whose presence signals a trend towards higher or lower values of such a continuous measure. While less

heavily investigated than discriminative motif discovery, a few correlative algorithms have been proposed, including MOTIF REGRESSOR [Conlon *et al.*, 2003], which first applies the (non-correlative) MDScan algorithm [Liu *et al.*, 2002] to identify motifs in a subset of high-scoring sequences, then filters the motif set based on the predictive value of regression models based on the selected motifs. Another correlative algorithm, FIRE [Elemento *et al.*, 2007], iteratively optimizes the mutual information between sequence scores and occurrences of candidate motifs, starting from a set of the most informative 7-mer "seed" motifs. Both of these algorithms may be seen as applying correlational information (regression or mutual information, respectively) as a filter to select and refine a set of candidate motifs—MDScan output for MOTIF REGRESSOR or the set of all 7-mers for FIRE—initialized in a non-correlative manner.

In contrast, we aimed to develop an algorithm for correlative motif discovery not requiring a prespecified set of seed motifs to minimize the possibility of missing informative motifs due to suboptimal seeding. Correlative motif discovery requires a rule for assigning numbers to motif patterns, which may then be correlated with sequence scores to rank and select motifs. The generation of a seed set of motif candidates paves the way for quantification by simply counting occurrences of the seed motifs within each of the sequenches w_b. However, as the number of potential motifs of length k grows exponentially with k, seeding in this manner generally requires either keeping the initial candidate motifs short (as FIRE does) or by using other (non-correlative) filters to prescreen candidate motifs (as done by MOTIF REGRESSOR).

The combinatorial explosion of possible motifs of length k also implies that, given a fixed set of sequences w_b to be analyzed, only a fraction of possible length-k motifs will actually be observed in any w_b for k suitably large. For example, given 1000 sequences each of length 1000, at most one million k-mers of any length k can be found. This suggests focusing only on substrings of the w_b instead of on the universe of possible patterns which could be present in the w_b. More specifically, we consider *suffixes*—substrings formed by deleting the beginning of a string—of the w_b, as these have two critical advantages. First, the number of suffixes of a set of sequences w_b grows linearly with (and is actually equal to) the combined length (i.e., $\sum_b |w_b|$), and second, lexicographical ordering allows us to assign each suffix a numerical value: the location of the suffix in the sorted list of all suffixes. Just as words sharing a common prefix are found close together in a dictionary, suffixes starting with a shared k-mer are assigned similar numeric values. SArKS develops this idea into a correlative motif discovery algorithm, with a natural extension for identification of longer multi-motif domains (MMDs) spanning tens to hundreds of bases, as described in Section 2.

We present the application of SArKS to real RNA-seq data, using nonparametric permutation testing to compute significance thresholds and estimating false positive rates. We demonstrate that SArKS outperforms several existing algorithms at identifying correlative motifs in a pair of cross-validation testing scenarios. Further analysis of the top motif patterns thus identified reveals among them many particular variations on sequence elements known to have regulatory activity [Mathelier *et al.*, 2015, Elbarbary *et al.*, 2016]. By implementing a correlational approach to motif discovery, SArKS thus provides a step forward in taking full advantage of the differential expression information offered by RNA-sequencing experiments in the context of motif discovery.

2 Methods

Symbolic notation is described as it is introduced in the text, but is also systematically described in the glossary provided in Section S2.1 for convenience.

Given n sequences w_b (also referred to as words) with associated scores y_b, the essential steps of the motif discovery algorithm we propose, illustrated in Fig 1 and more rigorously described in Section 2.1, consist of:

1. concatenating all the sequences w_b into one supersequence x (see Eq (1) below);
2. constructing the suffix array $[s_i]$ of this supersequence (Eq (4)), where i indexes all suffixes of x sorted into lexicographic order;
3. mapping the suffix positions i back to the sequences w_{b_i} from which the beginnings of the associated suffixes are derived (Eq (5)); and finally
4. for each i, applying kernel smoothing to locally regress the sequence scores y_{b_j} on suffix positions j lexically near i (Eq (6)).

We thus encode the motif pattern corresponding to the first few characters of the suffix of x beginning at character s_i with the numerical suffix array index value i. Because i gives the position of a suffix in the lexicographically sorted list of suffixes of the concatenated supersequence x, multiple occurrences of a highly conserved motif—even if they derive from different sequences w_b—will be consolidated into a run i, $i+1$, ..., j of consecutive index values. Averaging together runs of $j-i$ consecutive scores by kernel smoothing using a kernel of width $j-i$ thus offers a way to compare the scores y_{b_i}, $y_{b_{i+1}}$, ..., y_{b_j} to the overall score distribution.

2.1 Motif selection

2.1.1 Concatenation

Concatenate all words w_b (each assumed to end in the line-terminator character \$ lexically prior to all other characters) to form word

$$x = w_0 * w_1 * \ldots * w_{n-1} \tag{1}$$

of length $l_n = |x| = \sum_b |w_b|$. Define also

$$l_b = \sum_{b'<b} |w_{b'}| \tag{2}$$

Thus $x[l_b, l_{b+1}) = w_b$; that is, the substring of the concatenated string starting at position l_b (inclusive) and ending immediately before position l_{b+1} (exclusive) is the sequence w_b (in this paper the first character of a string w is denoted $w[0]$, the second $w[1]$, etc.).

2.1.2 Suffix sorting

Lexically sort suffixes

$$x_s = x[s, |x|) \tag{3}$$

w_b:	D	NA	AN	DRNAY	AY
y_b:	0	4	0	3	0

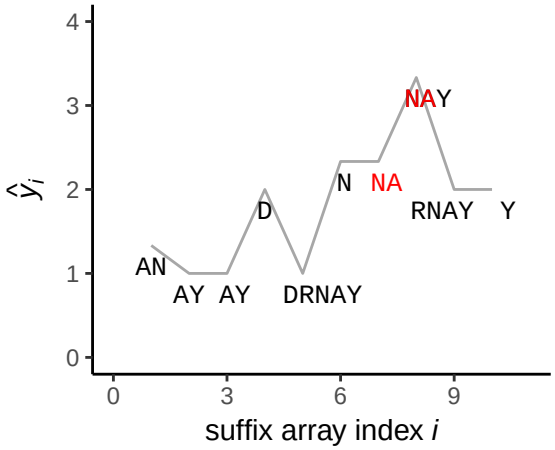
(A)

s	$x[s, \lvert x \rvert)$	b	y_b
0	D NA AN DRNAY AY	0	0
1	NA AN DRNAY AY	1	4
2	A AN DRNAY AY	1	4
3	AN DRNAY AY	2	0
4	N DRNAY AY	2	0
5	DRNAY AY	3	3
6	RNAY AY	3	3
7	NAY AY	3	3
8	AY AY	3	3
9	Y AY	3	3
10	AY	4	0
11	Y	4	0

(B)

i	s_i	$x[s_i, l_{b_i+1})$	b_i	y_{b_i}	$\hat{y}_i$
0	2	A	1	4	nan
1	3	AN	2	0	$4/3$
2	10	AY	4	0	1
3	8	AY	3	3	1
4	0	D	0	0	2
5	5	DRNAY	3	3	1
6	4	N	2	0	$7/3$
7	1	NA	1	4	$7/3$
8	7	NAY	3	3	$10/3$
9	6	RNAY	3	3	2
10	11	Y	4	0	2
11	9	Y	3	3	nan

(C)



(D)

Figure 1: **Overview of SArKS method. (A)** Concatenation of sequences w_b (end-of-sequence character indicated by white space instead of \$ for visual clarity) to form string $x =$ D\$NA\$AN\$DRNAY\$AY\$. **(B)** Table of all suffixes of x (part of each suffix following first end-of-sequence character shown in light gray), along with index b of input sequence w_b each suffix derived from and score y_b associated with w_b. **(C)** Sorted suffix table indicating suffix array index i, suffix array value s_i, suffix (sequence following first end-of-sequence character has been removed), sequence of origin b_i, associated score y_{b_i}, and smoothed score $\hat{y}_i$ generated using smoothing window of size 3 (kernel half-width $\kappa = 1$). **(D)** Smoothed scores $\hat{y}_i$ plotted against suffix array index i, indicating peak at $i = 8$ corresponding to suffix NAY of input sequence DRNAY. Note that prefix NA of this suffix is longest substring common to the two input sequences w_1 and w_3 with scores $y_b > 0$.

into ordered set

$$S = \{x_{s_0}, x_{s_1}, \cdots, x_{s_{l_n-1}}\} \tag{4}$$

thereby defining suffix array $[s_i]$ mapping index i of suffix in S to suffix position s in x (in our software we rely on the Skew algorithm [Kärkkäinen and Sanders, 2003] modified to use a difference cover of 7 and implemented in `SeqAn` [Döring *et al.*, 2008] to efficiently compute the suffix array).

2.1.3 Block marking

Define block array $[b_i]$ by

$$b_i = \max\{b \mid l_b \leq s_i\} \tag{5}$$

mapping index i of suffix in S to block b containing suffix position s_i. The block array then tells us that the character $x[s_i]$ at position s_i in the concatenated string x is derived from $w_{b_i}[s_i - l_{b_i}]$ in the sequence w_{b_i}.

2.1.4 Kernel smoothing

Calculate locally weighted averages

$$\hat{y}_i = \frac{\sum_j K_{ij}\, y_{b_j}}{\sum_j K_{ij}} \tag{6}$$

where the kernel K_{ij} acts as a weighting factor for the contribution of the score y_{b_j} to the smoothing window centered at sorted suffix index i. Loosely speaking, K_{ij} is used to measure how similar (the beginning of) the suffix $x[s_j, |x|)$ is to be considered to (the beginning of) the suffix $x[s_i, |x|)$ in the calculation of a representative score $\hat{y}_i$ averaged over suffixes similar to $x[s_i, |x|)$. As the suffixes have been sorted into lexicographic order, the magnitude of the difference $i - j$ provides some information regarding this similarity: the key idea of the kernel smoothing approach described here is that Eq (6) with K_{ij} defined to be a function of $|i - j|$ may therefore offer a computationally tractable approach for identifying similar substrings (prefixes of suffixes) which tend to occur preferentially in high scoring words w_b.

In this work we use a uniform kernel

$$K_{ij}^{(\kappa)} = \begin{cases} 1 & \text{if } |i-j| \leq \kappa \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

which allows Eq (6) to be computed easily in terms of cumulative sums:

$$\frac{\sum_j K_{ij}^{(\kappa)} y_{b_j}}{\sum_j K_{ij}^{(\kappa)}} = \frac{1}{2\kappa+1} \sum_{j=i-\kappa}^{i+\kappa} y_{b_j} = \frac{1}{2\kappa+1} \left(\sum_{j=1}^{i+\kappa} y_{b_j} - \sum_{j=1}^{i-\kappa-1} y_{b_j} \right) \tag{8}$$

The kernel half-width κ appearing in Eq (7) is an important adjustable parameter in the SArKS methodology controlling the degree of smoothing applied. Increasing κ smooths over more, and hence generally more diverse, suffixes, potentially increasing statistical power at the expense of the resolution of the detected motifs. We recommend investigating a range of values of this parameter as is illustrated in Section 3.

2.1.5 k selection

Set length $\hat{k}_i$ for k-mer associated with suffix array index i by locally averaging the length of suffix sequence identity:

$$\hat{k}_i = \frac{\sum_{j \neq i} K_{ij} \max \{k \leq k_{\max} \mid x[s_j, s_j + k) = x[s_i, s_i + k)\}}{\sum_{j \neq i} K_{ij}} \tag{9}$$

where $k_{\max}$ functions both to increase computational efficiency and to make $\hat{k}_i$ more robust in the presence of a small number of long identical substrings. All results presented here based on $k_{\max} = 12$: This value was selected as $k_{\max} \approx \log_4 |x|$ where x is the longest concatenated sequence string considered in Section 3.2, so that for $k > k_{\max}$ there are more distinct k-mers than there are positions for such k-mers to occur in all of the sequences w_b composing x.

Eq (9) is similar to Eq (6) except that: (a) Eq (9) smooths the length (capped at $k_{\max}$) of the longest prefix on which the suffixes $x[s_i, |x|)$ and $x[s_j, |x|)$ agree instead of smoothing the score y_{b_j} as in Eq (6); and (b) Eq (9) omits the central term $i = j$ as it trivially compares the suffix beginning at s_i to itself and is thus uninformative.

2.1.6 Motif selection

A straightforward approach to identifying correlative motifs using SArKS would then be to set a score threshold θ and take motifs to be k-mers prefixing the suffixes starting at the positions s_i in the concatenated string x. This is the essence of the method we propose, though with a couple of additional filters designed to pinpoint the optimal locations s_i at which initiate motifs and, in Section S2.2, to remove likely false positive positions.

Defining the negative spatial shift operator $\eta(i)$ which yields the unique suffix array index corresponding to the spatial position immediately prior to s_i, so that $s_{\eta(i)} = s_i - 1$, as well as the positive shift operator $\rho(i)$ similarly defined by the condition $s_{\rho(i)} = s_i + 1$, we start with a preliminary filtered suffix array index set I consisting of those i for which (1) the smoothed score $\hat{y}_i \geq \theta$ and (2) $\hat{y}_i$ is not less than the smoothed scores of the spatial positions in x immediately adjacent to s_i (i.e., s_i must be the loci of a peak in plot of $\hat{y}_i$ versus spatial position s_i):

$$I = \left\{ i \;\middle|\; (\hat{y}_i \geq \theta) \wedge \left(\hat{y}_{\eta(i)} \leq \hat{y}_i \geq \hat{y}_{\rho(i)} \right) \right\} \tag{10}$$

from which we obtain the associated set M of k-mers beginning at the positions s_i in x by

$$M = \left\{ x[s_i, s_i + \lfloor \hat{k}_i \rceil) \mid i \in I \right\} \tag{11}$$

where $\lfloor \hat{k}_i \rceil$ is the nearest integer to $\hat{k}_i$. Strategies for setting the filtering threshold θ based on the permutation testing method are described in Section S2.5. In the next section we recommend one additional filter be incorporated into the definition of the index set I, replacing Eq (10) by Eq (12).

2.2 Limit intra-sequence repeats

Short tandem repeats in the genome [Ellegren, 2004] can generate smoothing windows for the underlying repeated unit dominated by a relatively small number of distinct sequences, as is discussed in more detail in Section S2.2. As the scores for these motifs smooth over

fewer distinct sequence scores, the associated smoothed scores are less precise than those for most potential motifs, leading to increased false positive rates. Section S2.2 introduces the Gini impurity score g_i for suffix array index i measuring the "effective sequence count" contributing to the smoothing window centered at i, while Section S2.5 demonstrates that g_i predicts the variance of the smoothed score for suffix array index i under the null hypothesis of independence between sequence and score. Thus we modify equation (10) to remove likely false positive i values characterized by low Gini impurities g_i:

$$I = \left\{ i \;\middle|\; (\hat{y}_i \geq \theta) \wedge \left(\hat{y}_{\eta(i)} \leq \hat{y}_i \geq \hat{y}_{\rho(i)}\right) \wedge (g_i \geq g_{\min}) \right\} \tag{12}$$

screening out positions i for which the repeated occurrence of a few high-scoring words in the window centered at i leads to $\hat{y}_i \geq \theta$.

2.3 Spatial smoothing

Existing motif discovery approaches often take into account the tendency of regulatory motifs to cluster together [Wasserman and Sandelin, 2004]. Our algorithm builds on this observation, indentifying candidate regulatory regions through the application of a second round of kernel-smoothing over suffix positions s_i within words:

$$\hat{\hat{y}}_{s_i} = \frac{\sum_j L_{s_i t_j} \hat{y}_j}{\sum_t L_{s_i t}} \tag{13}$$

where we here use uniform kernels of the form

$$L^{(\lambda)}_{s_i t_j} = \begin{cases} 1 & \text{if } (0 \leq (t_j - s_i) < \lambda) \wedge (b_i = b_j) \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

(generally with width $\lambda \neq \kappa$) to search for regions of length λ with elevated densities of high-scoring motifs. Note that $\hat{\hat{y}}_{s_i}$ defined by Eq (13) is indexed not by suffix array index i but by suffix array value s_i giving the spatial position s_i in the concatenated word x.

To use such spatial smoothing for motif selection/filtering, it is necessary to introduce a second threshold θ_{spatial}, as the doubly-smoothed scores $\hat{\hat{y}}_{s_i}$ will generally be somewhat less dispersed than will be the singly-smoothed $\hat{y}_i$. The threshold θ_{spatial} can be used to define an index set I_{spatial} similar to the manner in which I is defined by Eq (12), as is detailed in Section S2.4.

Section S2.4 additionally defines the set J_{spatial} of suffix array indices i corresponding to the starting positions of MMDs. It then details the procedure adopted by SArKS to merge spatially contiguous motifs colocalized within the same MMD, yielding the set I_{spatial} of suffix array indices i and merged motif lengths $\hat{\hat{k}}_{s_i}$ required to obtain the merged motif set M_{spatial} analogous to equation (11).

2.4 Permutation testing

The significance of the observed correlation between the occurrence of the motifs uncovered by SArKS and the sequence scores y_b can be evaluated by examining results obtained when the sequences w_b and the scores y_b are independent of each other. To this end, the word scores y_b are subjected to permutation π to define

$$y_b^{(\pi)} = y_{\pi(b)} \tag{15}$$

If the permutation π is randomly selected independently of both the sequences w_b and the scores y_b, any true relationships between sequences and scores will be disrupted. Section S2.5 and Section S2.6 develop the strategy used by SArKS to set thresholds θ (and/or θ_{spatial}) for each particular combination of parameters $\kappa, \lambda, g_{\min}$ so as to control the overall false positive rate for the desired range of parameter sets.

3 Results and discussion

3.1 Illustration using simulated data

To illustrate SArKS, we first applied it to a simple simulated toy data set in which 30 random sequences w_b were generated with each letter $w_b[s]$ drawn independently from a Unif{A,C,G,T} distribution. We then embedded the k-mer motif CATACTGAGA ($k = 10$) in the last 10 sequences (i.e., those w_b with $b \geq 20$) by choosing a position s_b independently for each sequence w_b from Unif$\{0, \ldots, |w_b| - k\}$ and replacing $w_b[s_b, s_b + k)$ with the motif. Scores were assigned to the sequences according to whether the motif had been embedded:

$$y_b = \begin{cases} 0 & \text{if } b \in [0, 20) \\ 1 & \text{if } b \in [20, 30) \end{cases} \tag{16}$$

The kernel half-width $\kappa = 4$ was used to obtain smoothing windows of approximately the same size as the number of motif-positive sequences, $2\kappa + 1 \approx |\{b \mid y_b = 1\}|$. As this number will not be known in advance when applying SArKS to real data, we recommend testing a range of κ values as done in Section 3.2 below, but here it serves well for a simple demonstration.

Fig 2 plots $\hat{y}_i$ as obtained from Eq (6) applying the method of Section 2.1. The highest peaks in the plot correspond to the positions of various substrings of the embedded motif, and correspond to the set M of k-mers defined by the $x[s_i, s_i + \lfloor \hat{k}_i \rceil)$ column of Table 1.

Removing nested motifs from Table 1 as described in Section S2.3, Eq (S7) leaves only the rows for $i \in \{2257, 2258, 2256, 1462, 1458, 1463\}$. Applying Eq (S8) then extends the 8-mer ATACTGAG of the rows $i \in \{1462, 1458, 1463\}$ to the full 10-mer, so that, following Eq (S9), the final k-mer set $M' = \{\text{CATACTGAGA}\}$ is recovered.

Section S2.5 illustrates the utility of setting a minimum Gini impurity $g_{\min}$ during motif selection: 190 out of 1000 random permutations generated at least one position $i^{(\pi)}$ for which $\hat{y}^{(\pi)}_{i^{(\pi)}} = 1 \geq \theta$ (for this toy model θ was simply taken to have the maximum possible value of 1), but only 20 of these permutations yield any results if $g_{\min} = 1-(1+\gamma)(1-\operatorname{median}_i g_i) = 0.8506$ (following Eq (S5) with $\gamma = 0.1$) is applied. Based on these results, we can derive a 95% CI of (1.2%, 3.1%) for family-wise error rate (FWER).

3.2 DNA motifs associated with neuron subtype-specific expression

We also used SArKS to analyze RNA-seq gene expression data from distinct mouse neocortical neuron subtypes [Mo *et al.*, 2015]. Here, pooled genetically defined cell classes were subjected to a variety of sequencing methods to investigate transcriptomic and epigenetic variation, including cell class-specific ATAC-seq to determine what genomic regions are accessible to transcriptional machinery. Combined with the RNA-seq differential expression results, this

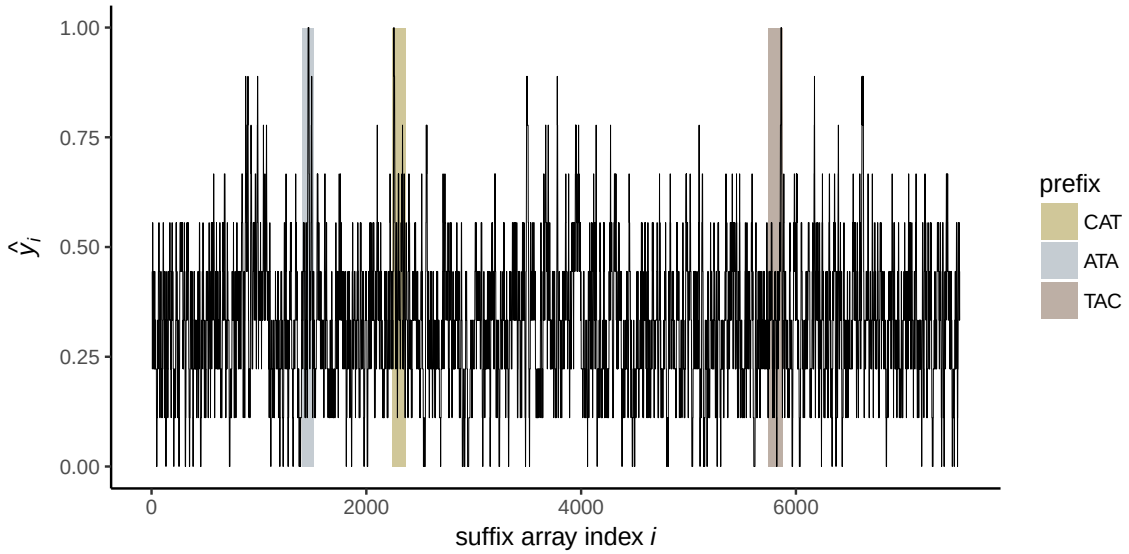


Figure 2: **Locating peaks in kernel-smoothed scores $\hat{y}_i$.** Variation of kernel-smoothed scores $\hat{y}_i$ (Eq (6), using kernel half-width $\kappa = 4$) along suffix array index i for simulated data set. Gold, silver, and bronze bars indicate positions in suffix array table of suffixes beginning with prefixes CAT, ATA, and TAC, which correspond to the first five characters of embedded motif CATACTGAGA. Detailed information on the peak locations at which the smoothed score $\hat{y}_i = 1$ is presented in table 1 below.

Table 1: **Suffix array peak positions with $\hat{y}_i \geq \theta$.**

i	s_i	$\hat{y}_i$	$\hat{k}_i$	$x[s_i, s_i + \lfloor \hat{k}_i \rceil)$	b_i	ω_i	g_i
2257	3959	1	10.25	CAT ACTGAGA	22	194	0.889
2258	4518	1	10.25	CAT ACTGAGA	25	0	0.889
2256	3544	1	9.62	CAT ACTGAGA	21	30	0.864
1460	3960	1	9.25	ATA CTGAGA	22	195	0.889
1461	4519	1	9.25	ATA CTGAGA	25	1	0.889
1459	3545	1	8.75	ATA CTGAGA	21	31	0.889
1462	3456	1	8.50	ATA CTGAG	20	193	0.864
1458	4442	1	8.25	ATA CTGAG	24	175	0.864
5864	3961	1	8.25	TAC TGAGA	22	196	0.889
5865	4520	1	8.25	TAC TGAGA	25	2	0.889
1463	5595	1	7.88	ATA CTGAG	29	73	0.864
5863	3546	1	7.75	TAC TGAGA	21	32	0.889
5862	4443	1	7.25	TAC TGAG	24	176	0.864
1464	5174	1	7.12	ATA CTGA	27	154	0.840
5861	5430	1	6.88	TAC TGAG	28	159	0.840
1465	4232	1	6.25	ATA CTG	23	216	0.815

Illustration of motif selection process from Section 2.1 applied to simulated data set using window half-width $\kappa = 4$. All positions for which the sequence smoothed score $\hat{y}_i \geq \theta = 1$ are shown; the table is sorted in descending order on the estimated motif length $\hat{k}_i$.

Table 2: **Filters applied to select gene set for SArKS analysis.**

Distinct Transcripts	Count
All	111,669
Detected	73,912
+ Highly Expressed	37,721
+ Highly Varying	29,164
- Duplicate Isoforms	11,857
+ Accessible	6,326

Number of distinct transcripts remaining after sequential application of filters described in Section S2.7.

information is useful for filtering the set of genes included in SArKS analysis, as discussed in Section S2.7.

Our goal was to identify potential regulatory motifs associated with transcripts enriched in parvalbumin (PV) GABAergic neurons. Because we focused on putative regulatory regions in the vicinity of actively used gene transcription start sites, we quantified expression at the transcript level, filtered transcripts, and determined differential expression as described in Section S2.7. Table 2 summarizes the results of the various transcript filters: 6,326 distinct transcripts, each representing a unique gene, were retained for analysis. For this data set, we conducted three separate SArKS analyses, two focusing upstream (5') of the TSSs for the transcripts of interest and the other downstream (3').

We examined upstream sequences w_b for each of the 6,326 remaining transcript species from -3,000 base pairs to the TSS (the transcription start sites of 85.5% of mouse genes annotated in Ensembl GRCm38 [Aken *et al.*, 2016] are separated by more than 3 kb; median separation is 23 kb; mean separation is 54 kb). We also performed separate analyses of sequences w_b extending from the TSS to +1000 base pairs after the TSS.

We tested a range of half-window sizes $\kappa \in \{250, 500, 1000, 2500\}$ with the maximum value of 2500 selected to produce a smoothed window of size $2\kappa + 1 = 5001$ similar to the number of sequences analyzed (6326). For each half-window size κ, we applied two minimum Gini impurity values $g_{\min}$ set according to Eq (S5) with first $\gamma = 0.1$ and then $\gamma = 0.2$. Also for each value of κ, we examined three separate values of the spatial window size $\lambda \in \{0, 10, 100\}$. These three values were selected to investigate the performance of SArKS using no spatial smoothing ($\lambda = 0$), using a window $\lambda = 10$ of the typical length scale of eukaryotic transcription factor binding sites [Stewart *et al.*, 2012], and finally using a window $\lambda = 100$ to target the low end of the enhancer length distribution [Loots, 2008].

3.2.1 Benchmark comparisons to existing algorithms

In order to contextualize the performance of SArKS, we conducted a cross-validation benchmarking study comparing the results of SArKS to those produced by four existing methods. Two of these methods, FIRE [Elemento *et al.*, 2007] and MOTIF REGRESSOR [Conlon *et al.*, 2003], were selected for comparison as they provide alternative approaches to correlative motif discovery as we have defined it here. The remaining two algorithms, DREME [Bailey, 2011] and HOMER [Heinz *et al.*, 2010], are popular discriminative methods which we have run by discretizing our score data with promoter sequences b considered 'positive' sequences

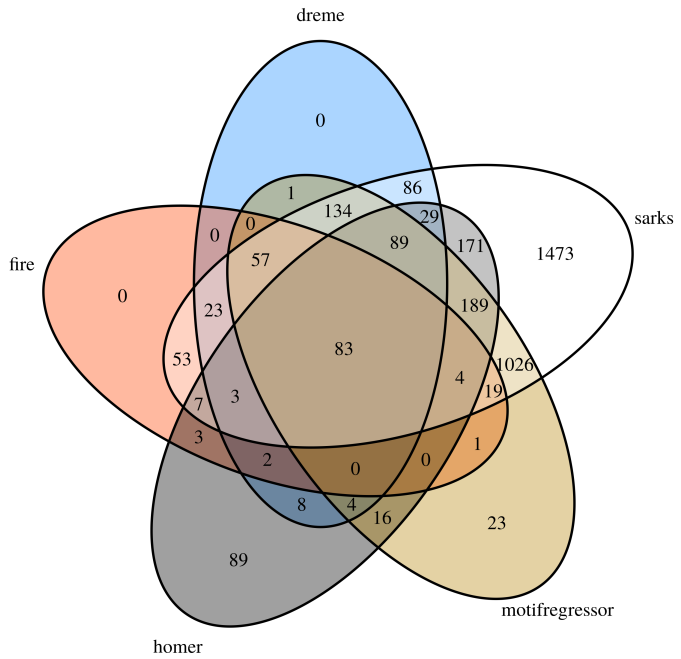


Figure 3: **Venn diagram of motif similarities shows substantial overlap between algorithms.** Venn diagram indicating the counts of motifs m (identified by one or more of the algorithms tested) for which all algorithms a in a particular subset $A \subset$ {dreme, fire, homer, motifregressor, sarks} identify a motif m'_a exhibiting significant `tomtom` similarity ($q \leq 0.1$) to m. SArKS identifies motifs with significant similarity to the majority of those detected by any of the algorithms.

if the score $y_b \geq 2$, 'negative' otherwise.

The set of 6,326 transcripts analyzed from the Mo 2015 data set was split into 5 disjoint subsets $V_1, V_2, \ldots, V_5$ (the name V_f intended to suggest the fth validation set) of approximately equal size ($|V_1| = 1,266$, while $|V_f| = 1,265$ for $f > 1$). For each of the algorithms evaluated and for both of the promoter sequence ranges investigated, 5 separate motif identification analyses were conducted by successively including only the pairs $\{(w_b, y_b) \mid b \notin V_f\}$, thus holding out those genes $b \in V_f$ for validation. Existing algorithms were run at their default parameter settings (defined either within the source code or in associated documentation; exact specifications are given in Section S2.8), with the exception that MOTIF REGRESSOR was run searching only for motifs positively correlating with score to enable more direct comparison of its output with that of the other algorithms (none of which look for anticorrrelated motifs, though all could do so simply by reversing the sign of the score).

In order to assess the degree of similarity in the motif sets identified by these methods, we used `tomtom` ([Gupta *et al.*, 2007]) to compare the pooled motif sets identified by each algorithm. Fig 3 shows the overlap between motifs sets by algorithm using this method of motif comparison: SArKS identifies similar motifs for the majority of motifs identified by each of the other algorithms, as well as many additional ones.

Given our interest in correlative motif discovery, the Pearson correlation between the count of occurrences of a given motif in sequence w_b with the score y_b across the sequence-score pairs (w_b, y_b) provides a natural metric for assessing motif discovery. Fig 4 plots the

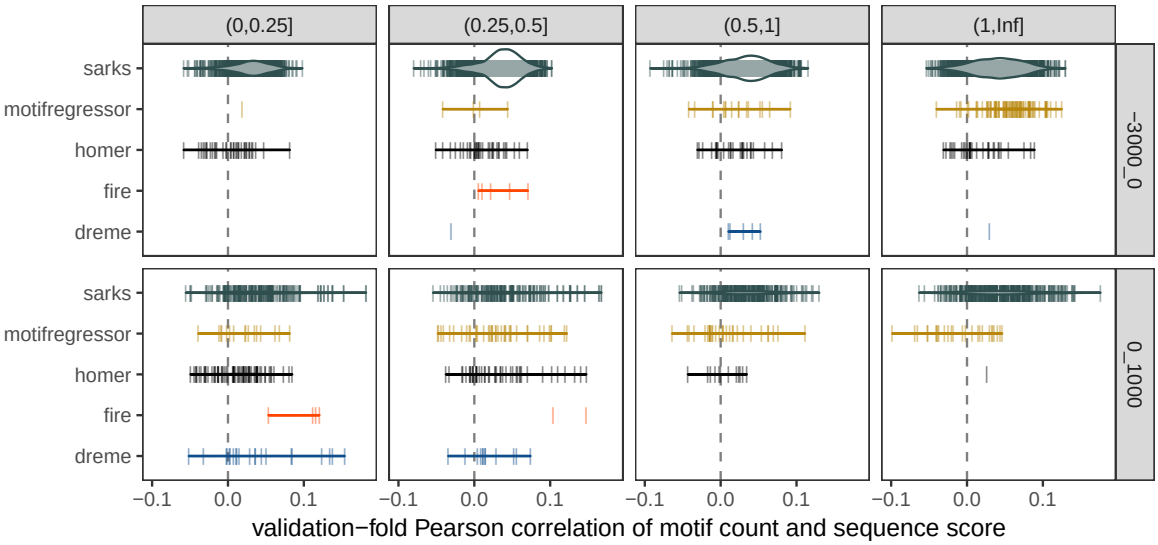


Figure 4: **Identified motifs correlate with gene specificity scores in held-out validation subsamples.** Each vertical line represents a motif identified by the indicated algorithm in one of the five cross-validation folds. The horizontal position of the line encodes the Pearson correlation coefficient of the motif count with the associated sequence score (calculated using only the genes in the held-out validation set for the cross-validation fold in which the motif was identified). Vertical panel: sequence region for motif counts, either 3kb upstream or 1kb downstream of TSS; horizontal panel: average number of occurrences of motif within sequence region across all analyzed genes. Widths of violins represent motif density and are scaled consistently across all panels.

estimated Pearson correlation coefficient values for each motif identified (by each algorithm) evaluated using the held-out validation set $\{(w_b, y_b) \mid b \in V_f\}$ appropriate for the fold f in which the motif was discovered. The count for a given motif in sequence w_b was assessed using `fimo` ([Grant *et al.*, 2011]) for DREME, HOMER, and MOTIF REGRESSOR—all of which represent motifs as position-weight matrices—and using a simple regular expression search for FIRE (which returns regular expression representations of motifs) and for SArKS k-mers. In all cases, motif counts were based on motif occurrences on either the forward or reverse strand. This is the default behavior for motif scanning with `fimo`; for regular expression searches, this was implemented by searching for occurrences of either a motif or its reverse complement in the forward strand.

As Fig 3 and Fig 4 demonstrate, the number of motifs identified by different algorithms is highly variable: DREME, FIRE $\ll$ HOMER, MOTIF REGRESSOR $<$ SArKS. The interpretation of the number of motifs is, however, complicated by two factors: (1) the occurrence rate of individual motifs in the relevant biological sequences (promoters, etc.) may differ substantially and (2) some motifs may be very similar in sequence and/or genomic location (e.g., which promoters contain them).

The first of these complications is illustrated in Fig 4 by facetting horizontally on motif occurrence rate (count per analyzed sequence): one visible trend here is that the HOMER-identified motifs tend to occur less frequently than do the MOTIF REGRESSOR motifs, indicating that HOMER tends to define motifs with more granularity than does MOTIF REGRESSOR. SArKS-identified motifs are spread across a wide range of per-sequence occurrence rates in this plot.

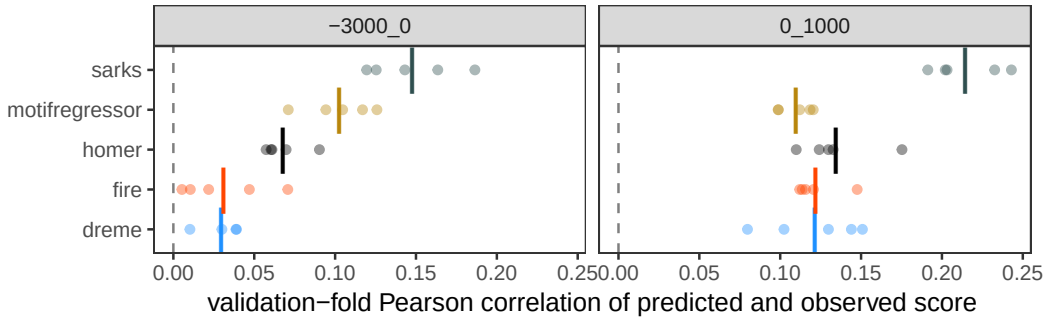


Figure 5: **Motif regression model predictions correlate with gene specificity scores in held-out validation subsamples.** Each of five cross-validation folds is plotted as separate point for each algorithm; fewer than five points are shown for corresponding algorithm if no motifs identified in one or more cross-validation folds. Left panel: regions 3kb upstream of TSS; right panel: regions 1kb downstream of TSS. Vertical lines indicate mean Pearson correlation across all folds for which value was calculable.

The second complication—the similarities among individual motifs identified by an algorithm—may be addressed by noting that correlative motif discovery can also be viewed as a form of feature extraction. In this vein, we can assess the performance of motif discovery algorithms by using the selected motifs as predictors to build a regression model for associated sequence score y_b based on the motif counts in the sequence w_b. Fig 5 plots the validation set-estimated Pearson correlation coefficients of the predictions made by building a linear ridge regression model (with the L2 regularization parameter selected by generalized cross-validation [Golub *et al.*, 1979]) with the sequence scores for each cross-validation fold by algorithm and sequence range. This approach collapses the variation in quantity and quality of individual motifs down to variation of a single quantity—the regression model predictions—thereby facilitating a head-to-head comparison of motif discovery algorithms bypassing both of the complications discussed above. (Note that because of the similarity of some identified motifs, which manifests as collinearity of predictors in a regression context, regularization is a key component of such a regression modeling approach.)

SArKS yields better results than the other algorithms for both the upstream and especially downstream sequence regions (Fig 5). Nonetheless, all five algorithms generally perform better when searching the downstream region. Using the downstream sequences as $\{w_b\}$, the four existing algorithms yield similar results with perhaps a slight edge to HOMER. With the upstream sequences instead provided as $\{w_b\}$, there is more separation between the existing algorithms, with the performance of the algorithms roughly in order of how many motifs they report (DREME, FIRE $<$ HOMER $<$ MOTIF REGRESSOR $<$ SArKS). The superior performance of motif discovery algorithms on downstream sequences over upstream ones coincides with the existence downstream of a particular family of very similar motifs—Esrrb, Essra, Esrrg from the JASPAR database—which is identified by each of the algorithms in the some form. In particular, variations on this motif dominate the downstream SArKS motif results, as is discussed further in Section 3.2.2.

An alternative approach to assessing motif discovery efforts is to compare the motifs identified algorithmically to databases of known TF-binding motifs, such as JASPAR [Mathelier *et al.*, 2015]. This is explored further in Section S3.2.1.

Table 3: **Utilized Gini index filters remove small fractions of suffix array positions.**

sequence range	γ	half-window κ	$g_{\min}$	fraction $g_i \geq g_{\min}$
Upstream	0.1	250	0.9976	92.44%
Upstream	0.1	500	0.9987	90.16%
Upstream	0.1	1000	0.9992	87.46%
Upstream	0.1	2500	0.9996	83.89%
Upstream	0.2	250	0.9974	95.27%
Upstream	0.2	500	0.9986	93.82%
Upstream	0.2	1000	0.9991	92.27%
Upstream	0.2	2500	0.9995	89.61%
Downstream	0.1	250	0.9976	94.12%
Downstream	0.1	500	0.9987	91.12%
Downstream	0.1	1000	0.9992	86.83%
Downstream	0.1	2500	0.9996	81.96%
Downstream	0.2	250	0.9974	97.04%
Downstream	0.2	500	0.9986	95.74%
Downstream	0.2	1000	0.9991	93.87%
Downstream	0.2	2500	0.9995	90.88%

Fraction of suffix array positions i for which Gini impurity values $g_i \geq g_{\min}$, with $g_{\min}$ selected according to Eq (S5).

3.2.2 Analysis of SArKS results

We also investigated the results of SArKS applied directly to the full set of 6,326 genes remaining after application of the filters detailed in Table 2 (i.e., not restricted to a particular cross-validation training set). Just as we did for the cross-validation approach described in Section 3.2.1, we investigated all combinations $(\kappa, \lambda) \in \{250, 500, 1000, 2500\} \times \{0, 10, 100\}$ for the smoothing half-width κ and spatial length λ. We again computed two values of $g_{\min}$ for each value of κ following Eq (S5) both with $\gamma = 0.1$ and $\gamma = 0.2$. The values of $g_{\min}$, along with the fraction of suffix array index values i for which $g_i \geq g_{\min}$, are listed in Table 3.

The analyses performed using the stricter $g_{\min}$ values obtained using $\gamma = 0.1$ yielded larger k-mer motif sets: 3,393 total distinct k-mers using $\gamma = 0.1$ versus only 1,232 $\gamma = 0.2$ for the upstream sequence set; 380 distinct k-mers using $\gamma = 0.1$ versus just 180 for the downstream sequence set. More than 98% of the k-mers discovered using $\gamma = 0.2$ were also identified using $\gamma = 0.1$ (for both sequence ranges: 1,208 of the 1,232 upstream; 179 of the 180 downstream). By demonstrating that more restrictive values of γ can yield larger motif sets that include almost all of the motifs obtained using more permissive γ values, these results highlight the importance of the Gini impurity filter in focusing SArKS on potential motifs that appear within sufficiently many distinct sequences w_b to achieve reasonable statistical confidence.

We assessed the statistical significance of these SArKS results following the method of Section S2.6 with thresholds θ and θ_{spatial} set by Eq (S24) and Eq (S25) using $z = 4$. Upstream sequence analysis considering 250 independent random permutations resulted

in 12 (4.8%) for which any of the parameter sets $(\kappa, g_{\min}, \lambda)$ yielded a nonempty set of identified motifs. These upstream sequence results correspond to a 95% family-wise error rate confidence interval (FWER CI) of (2.5%, 8.2%). For the downstream sequence analysis, 250 independent permutations yielded eight (3.2%) instances of nonempty motif sets, from which we estimate a 95% FWER CI of (1.4%, 6.2%).

The role of the parameter z in Eqs (S24)-(S25) in balancing FWER against sensitivity can be seen in the analyses presented here by considering the consequence of increasing z: at $z = 5$, only two of the 250 upstream permutations (0.8%) resulted in nonempty motif k-mer sets (95% FWER CI (0.097%, 2.9%)) while four of the 250 downstream permutations (1.6%) yielded any motif results (95% FWER CI (0.44%, 4.0%)). The cost of this decreased false positive rate to sensitivity is apparent in that only 49% of the upstream and 50% of the downstream motif k-mers identified using $z = 4$ are still discovered using $z = 5$. Here we are willing to accept the FWER values associated with $z = 4$ (point estimates 4.8% and 3.2% for upstream and downstream analyses, respectively) in order to maximize sensitivity by casting a reasonably wide net.

The highest $\hat{y}_i$ obtained in any of the SArKS analyses conducted—detected in the downstream sequence analysis using $\kappa = 250$, $\lambda = 0$, and $\gamma = 1.1$ in the downstream region analysis—corresponded to the k-mer TGACCTTG. This k-mer is very similar to a number of JASPAR TF-binding motifs. The strongest matches are to the binding motifs of Esrrb ($q = 0.00078$), Esrra ($q = 0.00078$), and Esrrg ($q = 0.00301$). In benchmarking SArKS against four existing motif-discovery algorithms, we noted that all of these algorithms identified motifs similar to each of these JASPAR motifs (`tomtom` $q \leq 0.1$). In the case of SArKS specifically, we find that in fact a large fraction of the motifs associated with identified peaks in $\hat{y}_i$ exhibit significant similarity to one of the JASPAR motifs Esrrb, Esrra, or Esrrg, as is illustrated in Fig 6B. This particular set of motifs may also help to explain the overall stronger performance of all of the motif discovery algorithms using the downstream sequences relative to the upstream sequences in the regression comparison (Fig 5), as was briefly alluded to in Section 3.2.1.

Esrrb/Essra/Esrrg binding motifs were identified by SArKS analysis of the upstream sequences, but they did not account for either the highest scores $\hat{y}_i$ nor did they correspond to a large fraction of the overall k-mer motif sets discovered—see Fig 6A. For five of the 12 distinct combinations of smoothing half-window κ and spatial window λ investigated using SArKS, the k-mer CCACCTGC was identified at the positions s_i with maximal values of $\hat{y}_i$ (the k-mers GCACACCTT, TGGAACTCACT, CCTGGAAC, and CAGCCTGG (identified using two distinct parameter combinations at the same suffix index i) were associated the highest $\hat{y}_i$ values using the remaining seven parameter combinations). The octamer CCACCTGC contains the canonical core recognition E-box sequence CANNTG (more specifically, the E12-box variant CACCTG [Bouard *et al.*, 2016]). Comparison of CCACCTGC with known motifs from the JASPAR database [Mathelier *et al.*, 2015] using `tomtom` finds some evidence of similarity to TF-binding motifs for SNAI2, MAX, SCRT2, SCRT1, TCF3, MNT, Id2, MAX::MYC, TCF4, and FIGLA (q-values of 0.14 for each), though no similarities were significant at $q \leq 0.1$.

As CCACCTGC was identified in analyses with spatial window length λ ranging up to 100, we performed a multiple sequence alignment using `muscle` [Edgar, 2004] of the 100-mers $x[s_i - 50, s_i + 50)$ for these positions s_i (Fig 7A); three of the five 100-mers thus aligned were very similar (Levenshtein distance ≤ 7) to the 99-mer consensus sequence constructed.

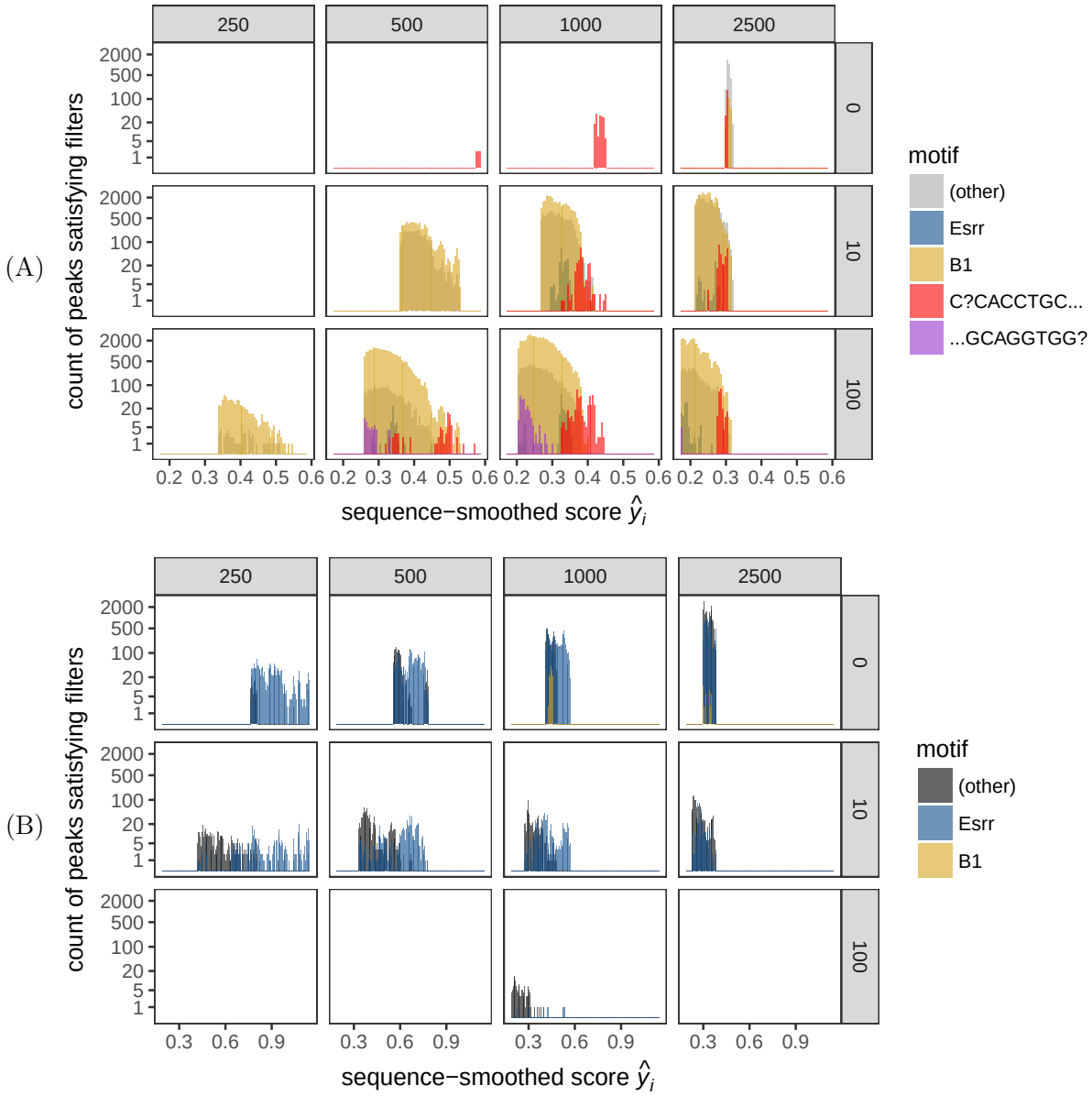


Figure 6: **Contributions of top motifs to peak composition. (A)** Log-scaled histograms of peaks $i \in I$ (or I_{spatial} when spatial smoothing is employed) identified in **upstream** analysis for which corresponding k-mer motifs: (1) are prefixed with CACCTGC or CCACCTGC (indicated in red) or are suffixed by the reverse complement sequences GCAGGTG or GCAGGTGG (purple); (2) are otherwise spatially located within a `blast` hit to the B1 SINE sequence (gold); (3) exhibited significant tomtom similarity ($q \leq 0.1$) to one of the JASPAR motifs Esrra, Esrrb, or Esrrg (blue); or (4) did not satisfy any of the above criteria (gray). Horizontal panels: half-window κ values used in analysis; vertical panels: spatial smoothing length λ. **(B)** Log-scaled histograms of peaks identified in **downstream** analysis; color coding is as in (A) except that black replaces gray. C?CACCTGC and its reverse complement do not occur in downstream peak set.

Furthermore, the consensus sequence also contains CCTGGAAC and CCAGGCTG (reverse complement of CAGCCTGG).

Interestingly, a screen of known repeated elements in the mouse genome for a consensus sequence uncovered a 93.9% identical base pair stretch of the B1 short interspersed element (SINE) sequence (SINEBase [Vassetzky and Kramerov, 2013]). The B1 SINE family consists of retrotransposon-derived sequences that appear throughout the mouse genome, especially upstream and within introns of genes implicated in DNA remodeling and expression regulation [Tsirigos and Rigoutsos, 2009]. Additional observations have been made further suggesting that SINE sequences function as transcriptional enhancers [Ichiyanagi, 2013, Elbarbary *et al.*, 2016, Ge, 2017].

Fig 6A indicates the number of SArKS-identified peaks that fall within `blast` hits between the upstream sequences w_b and the B1 SINE consensus sequence (minimum 90% sequence identity and 70 base alignment length) in gold, red, or purple. Those within-B1 peaks which corresponded to the flagged k-mers CCACCTGC or CACCTGC are coded as red, while those corresponding the reverse complements GCAGGTGG or GCAGGTG are coded purple; all remaining within-B1 peaks are indicated by gold. Fig 7B provides a more detailed view of these peak counts by splitting them out by position to which the corresponding k-mers align to the B1 consensus. Whether or not each base of each k-mer is a match or mismatch to the B1 consensus is also indicated in Fig 7B; it is apparent that, while essentially the entirety of the B1 consensus is represented by identified k-mer motifs, there is more variation away from the consensus towards the left end and at a couple of isolated positions further in than along most of the length of B1.

The k-mer CCACCTGC itself is not quite a perfect match to the annotated B1, containing a single base substitution away from the octamer CCGCCTGC whose reverse complement GCAGGCGG is found at positions 49-56 of the SINEBase B1 sequence. This substitution is responsible for the peak at position 54 in the mismatch counts in Fig 7B—one of the few positions at which there are more mismatches than matches. It is also worth noting that this G to A substitution creates the above noted E-box sequence CANNTG, and that the unmodified octamer CCGCCTGC does not match any JASPAR motifs at $q \leq 0.5$, unlike the modified sequence.

One of the remaining top upstream k-mer motifs mentioned above, GCACACCTT, similarly matches the nonamer GCACGCCTT spanning positions 15-23 of the SINEBase B1 sequence, but with a single G to A substitution. The modified nonamer GCACACCTT identified by SArKS shows significant similarity (tied q values of 0.038) to the JASPAR motifs TBX21, EOMES, TBX15, TBX1, and TBX2, while the unmodified B1 nonamer GCACGCCTT again shows no similarity to any JASPAR motifs at $q \leq 0.5$. The last of the top upstream k-mer motifs, TGGAACTCACT, is a perfect match to the reverse complement of positions 89-99 of the SINEBase B1 sequence.

Finally, in order to illustrate how SArKS can be used to help select candidate regulatory regions for promoting specific expression patterns, we again constructed a ridge regression model based on the counts of SArKS-identified k-mer motifs. We applied the same modeling strategy as described in Section 3.2.1, again using generalized cross-validation [Golub *et al.*, 1979] to determine the L2 regularization parameter, to the promoter regions defined by the 3,000 base pairs immediately preceding the TSSs of each of the 6,326 distinct analyzed transcripts. Each distinct transcript was then assigned a score by resubstitution into the resulting regression fit. Table 4 shows a sequence of filters in which these regression scores

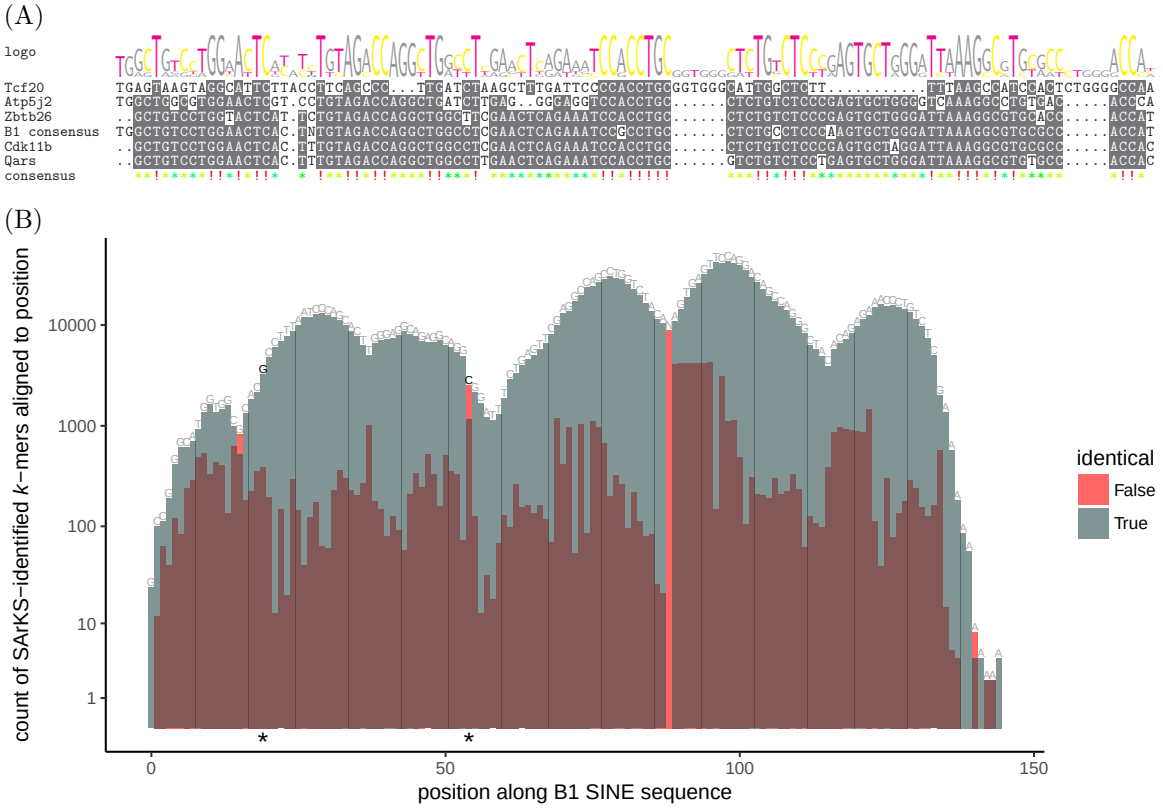


Figure 7: **SArKS-discovered motifs within B1 SINE elements. (A)** Multiple sequence alignment (`muscle`) of 100-mers surrounding top CCACCTGC motif peaks with reverse-complement of B1 consensus sequence. Associated genes are indicated to the left. Gray highlighting: $\geq 50\%$ agreement in the multiple sequence alignment. **(B)** Number of upstream motif k-mer peaks in B1 regions that align to each position within the B1 sequence. Gray bars: number of peak k-mers derived from upstream sequence regions for which a `blast` hit (percent identity $\geq 90\%$, alignment length ≥ 70) to B1 was found and for which an alignment of the k-mer to B1 aligned a matching base at the position in question. Red bars: number of k-mers within B1 blast hits which align aginst B1 with a *mismatched* base at the position in question. Above each bar is a label indicating the B1 consensus base at that position. Note that the lack of a gray bar at position 89 results from the lack of consensus base for B1 at this position (marked by N above the red bar), so that all k-mers that align against this position must produce a mismatch. The consensus base labels are drawn darker and the bars are marked with an asterisk at positions (19 and 54) where two of the top SArKS peaks exhibit changes compared to the B1 consensus sequence.

Table 4: **SArKS-based regression modeling assists in selecting candidate upstream regions for promoting PV-specific expression.**

Distinct Transcripts	Count
All analyzed	6,326
+ Protein coding	5,017
+ High expression in PV	1,595
+ Low expression in non-PV	196
+ PV : non-PV log-ratio ≥ 1	92
+ Top 5% SArKS regression score	13
+ Top 5% t-statistic score y_b	11

Number of distinct transcripts remaining after sequential application of described filters. Annotation of transcripts as protein coding or otherwise taken from Ensembl GRCm38 [Aken *et al.*, 2016]. Expression levels were considered high in PV samples if the average within-PV value of $\log_2(\text{TPM}+1) \geq \log_2(10+1)$, while expression levels were considered low in non-PV samples if the average non-PV $\log_2(\text{TPM}+1) < \log_2(10+1)$. Log-ratios were calculated as the difference of the PV-averaged- and non-PV-averaged-$\log_2(\text{TPM}+1)$ values, so that a log-ratio of one represents at least a two-fold increase in expression levels. SArKS regression scores were calculated using a ridge regression model built using counts of all k-mer motifs identified by SArKS applied to 3kb upstream promoter regions.

are applied alongside other relevant criteria to select candidate PV-specific promoter regions. The promoter regions associated with the genes ATP5SL, FAM195A, GPRC5B, IFT27, KCNH2, MAFB, PAQR4, PPFIA4, SLC29A2, SYT2, TBC1D2B, TMEM186, and TTC39A comprise the 11 candidates (from the final row of Table 4) selected for further experimental validation.

4 Conclusions

We introduce SArKS as a method for de novo correlative motif discovery and present examples of its application. This method fully exploits data sets produced by modern quantitative techniques, such as RNA-seq, by avoiding the dichotomization—and consequent loss of information [Fedorov *et al.*, 2009]—of sequence scores into discrete groups as required by discriminative motif discovery algorithms. SArKS has also been designed to minimize the reliance on specification of specific background sequence models, instead using nonparametric permutation methods [Ernst, 2004] to set significance thresholds for motif identification and estimate false positive rates. SArKS can also smooth over spatial location of motifs to identify multimotif domains (MMDs), which can then provide top-down feedback to enhance understanding of the motifs they contain. We have benchmarked SArKS against several existing discriminative and correlative motif discovery algorithms using a previously published RNA-seq data set: SArKS identifies particularly large bodies of motifs in these comparisons and SArKS-identified motif sets provide for more predictive feature sets in a regression modeling approach evaluated by cross-validation than do the motif sets generated by existing algorithms. Finally, by more closely examining the SArKS analysis results applied to this data set, we show that among the MMDs identified are many variations of the B1 short interspersed element (SINE) sequence in mouse, which has been previously implicated in gene regulation [Ichiyanagi, 2013, Elbarbary *et al.*, 2016, Ge, 2017].

References

[Aken *et al.*, 2016] Aken,B.L., Ayling,S., Barrell,D., Clarke,L., Curwen,V., Fairley,S., Fernandez Banet,J., Billis,K., García Girón,C., Hourlier,T. *et al.* (2016) The Ensembl gene annotation system. *Database,* **2016**.

[Badis *et al.*, 2009] Badis,G., Berger,M.F., Philippakis,A.A., Talukder,S., Gehrke,A.R., Jaeger,S.A., Chan,E.T., Metzler,G., Vedenko,A., Chen,X. *et al.* (2009) Diversity and complexity in DNA recognition by transcription factors. *Science,* **324** (5935), 1720–1723.

[Bailey, 2011] Bailey,T.L. (2011) DREME: motif discovery in transcription factor ChIP-seq data. *Bioinformatics,* **27** (12), 1653–1659.

[Bailey *et al.*, 2009] Bailey,T.L., Boden,M., Buske,F.A., Frith,M., Grant,C.E., Clementi,L., Ren,J., Li,W.W. and Noble,W.S. (2009) MEME SUITE: tools for motif discovery and searching. *Nucleic Acids Research,* , gkp335.

[Bailey and Elkan, 1995] Bailey,T.L. and Elkan,C. (1995) Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning,* **21** (1-2), 51–80.

[Bailey *et al.*, 2006] Bailey,T.L., Williams,N., Misleh,C. and Li,W.W. (2006) MEME: discovering and analyzing DNA and protein sequence motifs. *Nucleic Acids Research,* **34** (suppl 2), W369–W373.

[Barrett *et al.*, 2013] Barrett,T., Wilhite,S.E., Ledoux,P., Evangelista,C., Kim,I.F., Tomashevsky,M., Marshall,K.A., Phillippy,K.H., Sherman,P.M., Holko,M. *et al.* (2013) NCBI GEO: archive for functional genomics data sets—update. *Nucleic Acids Research,* **41** (D1), D991–D995.

[Bouard *et al.*, 2016] Bouard,C., Terreux,R., Honorat,M., Manship,B., Ansieau,S., Vigneron,A.M., Puisieux,A. and Payen,L. (2016) Deciphering the molecular mechanisms underlying the binding of the TWIST1/E12 complex to regulatory E-box sequences. *Nucleic Acids Research,* , gkw334.

[Bourgon *et al.*, 2010] Bourgon,R., Gentleman,R. and Huber,W. (2010) Independent filtering increases detection power for high-throughput experiments. *Proceedings of the National Academy of Sciences,* **107** (21), 9546–9551.

[Bray *et al.*, 2015] Bray,N., Pimentel,H., Melsted,P. and Pachter,L. (2015) Near-optimal RNA-Seq quantification. *arXiv preprint arXiv:1505.02710,* .

[Breiman *et al.*, 1984] Breiman,L., Friedman,J., Stone,C.J. and Olshen,R.A. (1984) *Classification and Regression Trees.* CRC press.

[Conlon *et al.*, 2003] Conlon,E.M., Liu,X.S., Lieb,J.D. and Liu,J.S. (2003) Integrating regulatory motif discovery and genome-wide expression analysis. *Proceedings of the National Academy of Sciences,* **100** (6), 3339–3344.

[Döring *et al.*, 2008] Döring,A., Weese,D., Rausch,T. and Reinert,K. (2008) SeqAn an efficient, generic C++ library for sequence analysis. *BMC Bioinformatics,* **9** (1), 11.

[Edgar, 2004] Edgar,R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research,* **32** (5), 1792–1797.

[Elbarbary *et al.*, 2016] Elbarbary,R.A., Lucas,B.A. and Maquat,L.E. (2016) Retrotransposons as regulators of gene expression. *Science,* **351** (6274), aac7247.

[Elemento *et al.*, 2007] Elemento,O., Slonim,N. and Tavazoie,S. (2007) A universal framework for regulatory element discovery across all genomes and data types. *Molecular Cell,* **28** (2), 337–350.

[Ellegren, 2004] Ellegren,H. (2004) Microsatellites: simple sequences with complex evolution. *Nature Reviews Genetics,* **5** (6), 435–445.

[Ernst, 2004] Ernst,M.D. (2004) Permutation methods: a basis for exact inference. *Statistical Science,* **19** (4), 676–685.

[Farcomeni, 2008] Farcomeni,A. (2008) A review of modern multiple hypothesis testing, with particular attention to the false discovery proportion. *Statistical Methods in Medical Research,* **17** (4), 347–388.

[Fauteux *et al.*, 2008] Fauteux,F., Blanchette,M. and Strömvik,M.V. (2008) Seeder: discriminative seeding DNA motif discovery. *Bioinformatics,* **24** (20), 2303–2307.

[Fedorov *et al.*, 2009] Fedorov,V., Mannino,F. and Zhang,R. (2009) Consequences of dichotomization. *Pharmaceutical Statistics,* **8** (1), 50–61.

[Fu, 2011] Fu,T.c. (2011) A review on time series data mining. *Engineering Applications of Artificial Intelligence,* **24** (1), 164–181.

[Ge, 2017] Ge,S.X. (2017) Exploratory bioinformatics investigation reveals importance of "junk" DNA in early embryo development. *BMC Genomics,* **18** (1), 200.

[Golub *et al.*, 1979] Golub,G.H., Heath,M. and Wahba,G. (1979) Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics,* **21** (2), 215–223.

[Gotea *et al.*, 2010] Gotea,V., Visel,A., Westlund,J.M., Nobrega,M.A., Pennacchio,L.A. and Ovcharenko,I. (2010) Homotypic clusters of transcription factor binding sites are a key component of human promoters and enhancers. *Genome Research,* **20** (5), 565–577.

[Grant *et al.*, 2011] Grant,C.E., Bailey,T.L. and Noble,W.S. (2011) FIMO: scanning for occurrences of a given motif. *Bioinformatics,* **27** (7), 1017–1018.

[Gupta *et al.*, 2007] Gupta,S., Stamatoyannopoulos,J.A., Bailey,T.L. and Noble,W.S. (2007) Quantifying similarity between motifs. *Genome Biology,* **8** (2), 1.

[Heinz *et al.*, 2010] Heinz,S., Benner,C., Spann,N., Bertolino,E., Lin,Y.C., Laslo,P., Cheng,J.X., Murre,C., Singh,H. and Glass,C.K. (2010) Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. *Molecular Cell,* **38** (4), 576–589.

[Huggins *et al.*, 2011] Huggins,P., Zhong,S., Shiff,I., Beckerman,R., Laptenko,O., Prives,C., Schulz,M.H., Simon,I. and Bar-Joseph,Z. (2011) DECOD: fast and accurate discriminative DNA motif finding. *Bioinformatics,* **27** (17), 2361–2367.

[Ichiyanagi, 2013] Ichiyanagi,K. (2013) Epigenetic regulation of transcription and possible functions of mammalian short interspersed elements, SINEs. *Genes & Genetic Systems,* **88** (1), 19–29.

[Kärkkäinen and Sanders, 2003] Kärkkäinen,J. and Sanders,P. (2003) Simple linear work suffix array construction. In *International Colloquium on Automata, Languages, and Programming* pp. 943–955 Springer.

[Liu *et al.*, 2002] Liu,X.S., Brutlag,D.L. and Liu,J.S. (2002) An algorithm for finding protein–DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments. *Nature Biotechnology,* **20** (8), 835–839.

[Long *et al.*, 2013] Long,H.K., Sims,D., Heger,A., Blackledge,N.P., Kutter,C., Wright,M.L., Grützner,F., Odom,D.T., Patient,R., Ponting,C.P. *et al.* (2013) Epigenetic conservation at gene regulatory elements revealed by non-methylated DNA profiling in seven vertebrates. *Elife,* **2**, e00348.

[Loots, 2008] Loots,G.G. (2008) Genomic identification of regulatory elements by evolutionary sequence comparison and functional analysis. *Advances in Genetics,* **61**, 269–293.

[Marsan and Sagot, 2000] Marsan,L. and Sagot,M.F. (2000) Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. *Journal of Computational Biology,* **7** (3-4), 345–362.

[Masoudi-Nejad *et al.*, 2012] Masoudi-Nejad,A., Schreiber,F. and Kashani,Z. (2012) Building blocks of biological networks: a review on major network motif discovery algorithms. *IET Systems Biology,* **6** (5), 164–174.

[Maston *et al.*, 2006] Maston,G.A., Evans,S.K. and Green,M.R. (2006) Transcriptional regulatory elements in the human genome. *Annual Review of Genomics and Human Genetics,* **7**, 29–59.

[Mathelier *et al.*, 2015] Mathelier,A., Fornes,O., Arenillas,D.J., Chen,C.y., Denay,G., Lee,J., Shi,W., Shyr,C., Tan,G., Worsley-Hunt,R. *et al.* (2015) JASPAR 2016: a major expansion and update of the open-access database of transcription factor binding profiles. *Nucleic Acids Research,* , gkv1176.

[Mo *et al.*, 2015] Mo,A., Mukamel,E.A., Davis,F.P., Luo,C., Henry,G.L., Picard,S., Urich,M.A., Nery,J.R., Sejnowski,T.J., Lister,R. *et al.* (2015) Epigenomic signatures of neuronal diversity in the mammalian brain. *Neuron,* **86** (6), 1369–1384.

[Nguyen and D'haeseleer, 2006] Nguyen,D.H. and D'haeseleer,P. (2006) Deciphering principles of transcription regulation in eukaryotic genomes. *Molecular Systems Biology,* **2** (1).

[Pavesi *et al.*, 2001] Pavesi,G., Mauri,G. and Pesole,G. (2001) An algorithm for finding signals of unknown length in DNA sequences. *Bioinformatics,* **17** (suppl 1), S207–S214.

[Pavesi *et al.*, 2004] Pavesi,G., Mereghetti,P., Mauri,G. and Pesole,G. (2004) Weeder Web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes. *Nucleic Acids Research,* **32** (suppl 2), W199–W203.

[Redhead and Bailey, 2007] Redhead,E. and Bailey,T.L. (2007) Discriminative motif discovery in DNA and protein sequences using the DEME algorithm. *BMC Bioinformatics,* **8** (1), 1.

[Sagot, 1998] Sagot,M.F. (1998) Spelling approximate repeated or common motifs using a suffix tree. In *Latin American Symposium on Theoretical Informatics* pp. 374–390 Springer.

[Sanyal *et al.*, 2012] Sanyal,A., Lajoie,B.R., Jain,G. and Dekker,J. (2012) The long-range interaction landscape of gene promoters. *Nature,* **489** (7414), 109–113.

[Segal *et al.*, 2002] Segal,E., Barash,Y., Simon,I., Friedman,N. and Koller,D. (2002) From promoter sequence to expression: a probabilistic framework. In *Proceedings of the sixth annual international conference on Computational Biology* pp. 263–272 acm.

[Segal and Sharan, 2005] Segal,E. and Sharan,R. (2005) A discriminative model for identifying spatial cis-regulatory modules. *Journal of Computational Biology,* **12** (6), 822–834.

[Sinha, 2003] Sinha,S. (2003) Discriminative motifs. *Journal of Computational Biology,* **10** (3-4), 599–615.

[Smyth, 2004] Smyth,G.K. (2004) Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology,* **3** (1), 1–25.

[Stewart *et al.*, 2012] Stewart,A.J., Hannenhalli,S. and Plotkin,J.B. (2012) Why transcription factor binding sites are ten nucleotides long. *Genetics,* **192** (3), 973–985.

[Tsirigos and Rigoutsos, 2009] Tsirigos,A. and Rigoutsos,I. (2009) Alu and B1 repeats have been selectively retained in the upstream and intronic regions of genes of specific functional classes. *PLoS Computational Biology,* **5** (12), e1000610.

[Valen *et al.*, 2009] Valen,E., Sandelin,A., Winther,O. and Krogh,A. (2009) Discovery of regulatory elements is improved by a discriminatory approach. *PLoS Computational Biology,* **5** (11), e1000562.

[Vassetzky and Kramerov, 2013] Vassetzky,N.S. and Kramerov,D.A. (2013) SINEBase: a database and tool for SINE analysis. *Nucleic Acids Research,* **41** (D1), D83–D89.

[Walhout, 2006] Walhout,A.J. (2006) Unraveling transcription regulatory networks by protein–DNA and protein–protein interaction mapping. *Genome Research,* **16** (12), 1445–1454.

[Wasserman and Sandelin, 2004] Wasserman,W.W. and Sandelin,A. (2004) Applied bioinformatics for the identification of regulatory elements. *Nature Reviews Genetics,* **5** (4), 276–287.

[Wunderlich and Mirny, 2009] Wunderlich,Z. and Mirny,L.A. (2009) Different gene regulation strategies revealed by analysis of binding motifs. *Trends in Genetics,* **25** (10), 434–440.

[Xie *et al.*, 2005] Xie,X., Lu,J., Kulbokas,E., Golub,T.R., Mootha,V., Lindblad-Toh,K., Lander,E.S. and Kellis,M. (2005) Systematic discovery of regulatory motifs in human promoters and 3 UTRs by comparison of several mammals. *Nature,* **434** (7031), 338–345.

[Yao *et al.*, 2014] Yao,Z., MacQuarrie,K.L., Fong,A.P., Tapscott,S.J., Ruzzo,W.L. and Gentleman,R.C. (2014) Discriminative motif analysis of high-throughput dataset. *Bioinformatics,* **30** (6), 775–783.

5 Acknowledgements:

The authors thank Becca Young, Eric Brenner, Brian Gereke and Preeti Mehta for helpful discussions and Ila Fiete for a critical reading of the manuscript. This project has been supported by NIH BRAIN Initiative award U01NS094330 to BVZ.

S2 Methods supplementary

S2.1 Notation glossary

$u * v$	concatenate strings u and v
$\|u\|$	length of string u
$u[i, j)$	substring of u starting at i^{th} character (inclusive) and continuing up until j^{th} character (exclusive) using 0-based indexing
$u[i, j]$	substring of u starting at i^{th} character (inclusive) and continuing through the jth character (inclusive) using 0-based indexing
$\{a \mid C(a)\}$	set containing all elements a satisfying condition $C(a)$
$\{a \mid C_1(a) \wedge C_2(a)\}$	set containing all elements a satisfying conditions $C_1(a)$ and $C_2(a)$
$\{a \mid C_1(a) \vee C_2(a)\}$	set containing all elements a satisfying conditions $C_1(a)$ or $C_2(a)$
$\mathbb{E}[A]$	expectation value of random variable A
$\mathbb{V}[A]$	variance of random variable A
i	suffix array index: 0-based position of suffix in lexicographically sorted list of all suffixes of string x
s_i	suffix array value: 0-based spatial position of suffix with suffix array index i within string x
b_i	block array value: 0-based position of block/word in which suffix with suffix array index i begins
$\hat{y}_i$	kernel smoothed score associated with suffix array index i
κ	half-width of kernel applied to generate $\hat{y}_i$
$\hat{k}_i$	estimate of smoothed k-mer length at suffix array index i
$\eta(i)$	negative spatial shift operator defined by property $s_{\eta(i)} = s_i - 1$
$\rho(i)$	positive spatial shift operator defined by property $s_{\rho(i)} = s_i + 1$
θ	threshold value for $\hat{y}_i$ for sequence-smoothed peak calling
I	set of suffix array indices identified as peaks by SArKS
M	set of k-mer motifs derived from suffix array peak set
$f_b^{(i)}$	weighted frequency of block/word b within smoothing window centered on suffix array index i
g_i	Gini impurity of smoothing window centered on suffix array index i
$g_{\min}$	minimum value of smoothing window Gini impurity for inclusion in peak set I
$\hat{\hat{y}}_{s_i}$	spatially smoothed score associated with spatial array value (spatial position) s_i
λ	length of spatial kernel applied to generate spatially smoothed scores $\hat{\hat{y}}_{s_i}$
$\hat{\hat{k}}_{s_i}$	estimate of merged k-mer length at spatial position (suffix array value) s_i
θ_{spatial}	threshold value for $\hat{\hat{y}}_{s_i}$ to call significant spatial windows
I_{spatial}	set of suffix array indices identified as k-mer starting positions using spatial smoothing
M_{spatial}	set of k-mer motifs derived from suffix array index set I_{spatial} using spatial smoothing
π	permutation of n blocks/words
Π	random variable representing randomly generated permutation
$\hat{y}_i^{(\pi)}$	sequence smoothed scores calculated with word scores permuted by π
$\hat{\hat{y}}_{s_i}^{(\pi)}$	spatially smoothed scores calculated with word scores permuted by π

S2.2 Limit intra-sequence repeats supplementary

One complicating factor in the strategy described in Section 2.1 is the presence of tandem repeats (common in eukaryotic DNA [Ellegren, 2004]): if the substring $x[s_i, s_i + rm)$ (assumed to derive wholly from the single word w_{b_i}) consists of $r \gg 1$ repeats of the same m-mer,

$$x[s_i, s_i + rm) = \underbrace{x[s_i, s_i + m)}_{1} * \underbrace{x[s_i, s_i + m)}_{2} * \cdots * \underbrace{x[s_i, s_i + m)}_{r} \tag{S1}$$

then it is likely that the sorted suffix array index positions j and k implicitly defined by $s_j = s_i + am$ and $s_k = s_i + bm$ for small $a, b \geq 0$ will be close by, since, assuming without loss of generality that $a < b$,

$$x[s_i + am, s_i + (r - b + a)m) = x[s_i + bm, s_i + rm) \tag{S2}$$

showing that the suffixes of x beginning at positions $(s_i + am)$ and $(s_i + bm)$ agree on their first $(r - b)m$ characters. Since all of the positions $s_i + am$ for small a must come from the same word block b_i they must have the same associated score y_{b_i}. If this score y_{b_i} is particularly high, this phenomenon may lead to windows of high $\hat{y}_j$ values centered on j satisfying $s_j = s_i + am$ which result from a very small number of different repeat-containing words (perhaps as few as one if the number of repeats is high enough within a single high-scoring word). We thus here develop a natural method for filtering the peak index set I to selectively remove suffix array index values i for which the smoothing window is dominated by a few heavily repeated words w_b.

The distribution of weighted word frequencies

$$f_b^{(i)} = \frac{\sum_j K_{ij} \delta_{b_j b}}{\sum_j K_{ij}} \tag{S3}$$

contributing to the window centered at position i of the suffix array table across the full word set W may for these purposes be summarized by the associated Gini impurity (often used in fitting classification and regression trees [Breiman *et al.*, 1984]):

$$g_i = \sum_b f_b^{(i)} \left(1 - f_b^{(i)}\right) \tag{S4}$$

which provides a measure ranging from 0 to $\frac{2\kappa}{2\kappa+1}$ of the degree of uniqueness of the words contributing to the calculation of $\hat{y}_i$.

As a concrete example, if all of the weighted frequencies word frequencies $f_b^{(i)} = \frac{1}{q}$ are the same for a set of exactly q words w_b appearing in the smoothing window centered on i, $g_i = 1 - \frac{1}{q}$. This suggests an intuitive interpretation of $(1 - g_i)$ as the multiplicative inverse of the "effective word count" contributing to the smoothing window around i.

Section S2.5 further demonstrates that $(1 - g_i)$ is also approximately proportional to the variation of the smoothed scores $\hat{y}_i$ that would be expected if there were no association between the sequences w_b and the scores y_b (see Eq (S21) below). This proportionality suggests a simple method for selection of a $g_{\min}$ value at which most suffix array indices i will be retained while filtering out only those most likely to yield false positive results under permutation testing:

$$1 - g_{\min} = (1 + \gamma) \left(1 - \operatorname*{median}_i g_i\right) \tag{S5}$$

As shown in Eq (S21), setting $g_{\min}$ to satisfy Eq (S5) removes suffix indices i for which the variance of the permuted smoothed scores is greater than $(1+\gamma)$ times the median value. Thus any value $\gamma > 0$ will retain the majority of positions i for further consideration. We have used $\gamma = 0.1$ or $\gamma = 0.2$ for all of the examples in the present work, retaining positions for which the permuted score variance is less than 110% or 120%, respectively, of the median.

Figure S1: **Example k-mers to be removed or extended.** Two identified k-mers (u=CATACTGAGA on the left and v=ATACTG on the right) are indicated by the dark gray highlighting, with two additional separately identified k-mers that are part of u indicated within the nested black boxes. The two nested k-mers contained within the boxes inside of u will be removed from the discovered k-mer set by the method of Section S2.3.1, while k-mer v = ATACTG will be extended by the method of Section S2.3.2 to include the characters highlighted in light gray, replacing v with CATACTGAGA. The four k-mers indicated in this figure correspond to positions $s_i \in \{3959, 3960, 3961, 4232\}$ from Section 3.1.

Requiring $g_i \geq g_{\min}$ results in redefining the peak index set I to

$$I = \left\{ i \;\middle|\; (\hat{y}_i \geq \theta) \wedge \left(\hat{y}_{\eta(i)} \leq \hat{y}_i \geq \hat{y}_{\rho(i)}\right) \wedge (g_i \geq g_{\min}) \right\} \tag{S6}$$

screening out positions i for which the repeated occurrence of a few high-scoring words in the window centered at i leads to $\hat{y}_i \geq \theta$.

S2.3 Removing and extending k-mers supplementary

The presence of a k-mer $x[s_i, s_i + k)$ associated with a high smoothed score $\hat{y}_i$ can also result in high smoothed scores $\hat{y}_j$ when $s_j = s_i + m$ if the substring $(k - m)$-mers $x[s_i + m, s_i + k)$ also also preferentially found in higher-scoring sequences, as pictured in Fig S1. The following two steps may be added to the algorithm described in Section 2.1 in order to reduce the reporting of such substrings when they are present only as part of the full k-mer:

S2.3.1 Removing nested k-mers

Cases in which both k-mer $x[s_i, s_i + k)$ and its sub-$(k - m_1 - m_2)$-mer $x[s_i + m_1, s_i + k - m_2)$ (with $m_1 > 0, m_2 \geq 0$) are individually identified can be resolved to report only the longer k-mer by removing any index $i \in I$ (defined by Eq (S6)) if there exists $j \in I$ such that the $\lfloor \hat{k}_j \rceil$-mer interval starting at s_j includes all of the $\lfloor \hat{k}_i \rceil$-mer interval starting at s_i, thus retaining only:

$$I' = \left\{ i \in I \;\middle|\; \forall j \in I : \; (s_i \leq s_j) \vee \left(s_i + \lfloor \hat{k}_i \rceil > s_j + \lfloor \hat{k}_j \rceil\right) \right\} \tag{S7}$$

This can be done efficiently using an interval tree.

S2.3.2 Extending substring k-mers

Fig S1 depicts not only two cases of nested k-mers which may be removed from the reported motif set by the method of Section S2.3.1 (contained within black boxes) but also an additional shorter k-mer ATACTG, highlighted in dark gray, which is derived from a distinct occurrence of the same longer k-mer (CATACTGAGA in this case). Because this distinct occurrence of the longer k-mer was not itself initially identified, the method of Section S2.3.1 does not remove the shorter substring k-mer from the motif set. However, such substring k-mers may be extended to the longer k-mer occurrence by the following method: for each $i \in I'$, define the duplet

$$\left(z_i^0, z_i^1\right) = \underset{z^0, z^1 \geq 0}{\arg\max} \left\{ z^0 + z^1 \;\middle|\; \exists j \in I' : \; x\left[s_i - z^0, s_i + \lfloor \hat{k}_i \rceil + z^1\right) = x\left[s_j, s_j + \lfloor \hat{k}_j \rceil\right) \right\} \tag{S8}$$

resolving any ties in the $\arg\max$ in favor of maximal z^0. Eq (S8) picks out the largest super-interval $\left[s_i - z^0, s_i + \lfloor \hat{k}_i \rceil + z^1\right)$ containing the interval $\left[s_i, s_i + \lfloor \hat{k}_i \rceil\right)$ such that the extended

$\left(\lfloor \hat{k}_i \rceil + z_i^0 + z_i^1\right)$-mer $x\left[s_i - z^0, s_i + \lfloor \hat{k}_i \rceil + z^1\right)$ is equal to one of the already identified k-mers $\left\{x\left[s_j, s_j + \lfloor \hat{k}_j \rceil\right) \,\middle|\, j \in I'\right\}$. (In the example of Fig S1, $z_i^0 = 1$ and $z_i^1 = 3$, corresponding to the light gray highlighted characters surrounding the substring k-mer). Then

$$M' = \left\{x\left[s_i - z_i^0, s_i + \lfloor \hat{k}_i \rceil + z_i^1\right) \,\middle|\, i \in I'\right\} \tag{S9}$$

defines our motif set after removal of nested motifs.

S2.4 Spatial smoothing supplementary

SArKS indentifies candidate multi-motif domains (MMDs) through the application of a second round of kernel-smoothing over suffix positions s_i within words:

$$\hat{\hat{y}}_{s_i} = \frac{\sum_j L_{s_i t_j}\, \hat{y}_j}{\sum_t L_{s_i t}} \tag{S10}$$

where we here use uniform kernels of the form

$$L_{s_i t_j}^{(\lambda)} = \begin{cases} 1 & \text{if } (0 \leq (t_j - s_i) < \lambda) \wedge (b_i = b_j) \\ 0 & \text{otherwise} \end{cases} \tag{S11}$$

(generally with width $\lambda \neq \kappa$) to search for regions of length λ with elevated densities of high-scoring motifs. Note that $\hat{\hat{y}}_{s_i}$ defined by Eq (S10) is indexed not by suffix array index i but by suffix array value s_i giving the spatial position s_i in the concatenated word x.

To use such spatial smoothing for motif selection/filtering, it is necessary to introduce a second threshold θ_{spatial}, as the doubly-smoothed scores $\hat{\hat{y}}_{s_i}$ will generally be somewhat less dispersed than will be the singly-smoothed $\hat{y}_i$. The threshold θ_{spatial} can be used to define an index set I_{spatial} similar to the manner in which I is defined by Eq (S6), but the task is more complex when we replace the single spatial position s_i by a spatial window $[s_i, s_i + \lambda)$.

Recalling the definitions of the negative/positive spatial shift operators $\eta(i)/\rho(i)$ which yield the unique suffix array indexes corresponding to the spatial position immediately before/after s_i, so that $s_{\eta(i)} = s_i - 1$ and $s_{\rho(i)} = s_i + 1$, first define:

$$J_{\text{spatial}} = \left\{i \,\middle|\, \left(\hat{\hat{y}}_{s_i} \geq \theta_{\text{spatial}}\right) \wedge (g_i \geq g_{\min}) \wedge \left(\hat{y}_{\eta(i)} \leq \hat{y}_i \geq \hat{y}_{\rho(i)}\right)\right\} \tag{S12}$$

J_{spatial} represents the set of suffix array indices i corresponding to the left endpoints s_i of spatial windows $[s_i, s_i + \lambda)$ passing the filters for score threshold θ_{spatial} and minimum Gini impurity $g_{\min}$, and for which the sequence-smoothed score $\hat{y}_i$ is at least as high as the spatially adjacent scores $\hat{y}_{\eta(i)}$ to the left and $\hat{y}_{\rho(i)}$ to the right.

Defining the left-directed distance δ_j from the suffix with sorted suffix array index j to the set J_{spatial} by

$$\delta_j = \min_{i \in J_{\text{spatial}}} \{s_j - s_i \mid s_i \leq s_j\} \tag{S13}$$

we define in turn the set of sorted suffix array indices I_{spatial} marking the starting positions of selected k-mers by:

$$I_{\text{spatial}} = \left\{i \,\middle|\, (\delta_i < \lambda) \wedge (\hat{y}_i \geq \theta_{\text{spatial}}) \wedge \left(\left(\delta_{\eta(i)} \geq \lambda\right) \vee \left(\hat{y}_{\eta(i)} < \theta_{\text{spatial}}\right)\right)\right\} \tag{S14}$$

Eq (S14) identifies suffix array indices i: (1) whose spatial positions s_i fall within a spatial window $[s_j, s_j + \lambda)$ for some $j \in J_{\text{spatial}}$, (2) whose sequence-smoothed score $\hat{y}_i \geq \theta_{\text{spatial}}$, and (3) for which the position $s_i - 1$ spatially to the left is either (3A) not in one of the spatial windows specified by J_{spatial} or (3B) has associated sequence-smoothed score $\hat{y}_{\eta(i)} < \theta_{\text{spatial}}$. This final criterion is

included because we want to merge adjacent k-mers whose leftmost positions fall within the same selected spatial window.

This merging process is implemented by calculating for each index $i \in I_{\text{spatial}}$ the length

$$\hat{\hat{k}}_{s_i} = \max_j \left\{ \hat{k}_j + s_j - s_i \;\middle|\; (\delta_j < \lambda) \wedge (s_m \in [s_i, s_j] \implies \hat{y}_m \geq \theta_{\text{spatial}}) \right\} \tag{S15}$$

of the merged k-mer starting at i. Eq (S15) sets $\hat{\hat{k}}_{s_i}$ by selecting the right endpoint $\hat{k}_j + s_j$ of the k-mer beginning at s_j to maximize the merged length $\hat{k}_j + s_j - s_i$ over all choices of s_j for which every position s_m between s_i and s_j has an acceptable sequence-smoothed score $\hat{y}_m \geq \theta_{\text{spatial}}$. It is then straightforward to obtain the motif set

$$M_{\text{spatial}} = \left\{ x\Big[s_i,\, s_i + \left\lfloor \hat{\hat{k}}_{s_i} \right\rfloor\Big) \;\middle|\; i \in I_{\text{spatial}} \right\} \tag{S16}$$

S2.5 Permutation testing supplementary

The significance of the observed correlation between the occurrence of the motifs uncovered by SArKS and the sequence scores y_b can be evaluated by examining results obtained when the sequences w_b and the scores y_b are independent of each other. To this end, the word scores y_b are subjected to permutation π to define

$$y_b^{(\pi)} = y_{\pi(b)} \tag{S17}$$

If the permutation π is randomly selected independently of both the sequences w_b and the scores y_b, any true relationships between sequences and scores will be disrupted. This suggests a simple method for assessing the significance of motifs discovered using a given set of parameters (kernel half-width κ, θ, $g_{\min}$, etc.): generate R random permutations π_r and for each permutation calculate scores $\hat{y}_i^{(\pi_r)}$ using Eq (6) (and also $\hat{\hat{y}}_{s_i}^{(\pi_r)}$ using Eq (S10) if spatial smoothing is employed) with y_b replaced by y_{π_b}. In this manner one can estimate the distribution of scores under a null model in which there is no association between the sequences of the various words w_b and the scores y_{π_b}.

This method of significance testing also provides the motivation for the form of Eq (S4) in Section S2.2. To demonstrate this, let Π be a random variable representing a random permutation and note that the random variables $y_{\Pi(b)}$ satisfy

$$\mathbb{E}\left[\hat{y}_i^{(\Pi)}\right] = \mathbb{E}\left[\frac{\sum_j K_{ij}\, y_{\Pi(b_j)}}{\sum_j K_{ij}}\right] = \frac{\sum_j K_{ij}\, \mathbb{E}\left[y_{\Pi(b_j)}\right]}{\sum_j K_{ij}} = \bar{y} \tag{S18}$$

while, assuming that the number of words $n = |W|$ is large enough that we may approximate $y_{\Pi(b)} \perp\!\!\!\perp y_{\Pi(b')}$ for $b \neq b'$,

$$\mathbb{V}\left[\hat{y}_i^{(\Pi)}\right] = \mathbb{V}\left[\frac{\sum_j K_{ij}\, y_{\Pi(b_j)}}{\sum_j K_{ij}}\right] \approx \sum_b \mathbb{V}\left[f_b^{(i)} y_{\Pi(b)}\right] = \mathbb{V}\left[y_{\Pi(\cdot)}\right] \sum_b \left[f_b^{(i)}\right]^2 \tag{S19}$$

where $f_b^{(i)}$ is defined by Eq (S3) and for all b

$$\mathbb{V}\left[y_{\Pi(\cdot)}\right] = \mathbb{V}\left[y_{\Pi(b)}\right] = \frac{1}{n} \sum_{b'} \left(y_{b'} - \bar{y}\right)^2 \tag{S20}$$

Eq (S19) then tells us that

$$\mathbb{V}\left[\hat{y}_i^{(\Pi)}\right] \propto \left[f_b^{(i)}\right]^2 = 1 - g_i \tag{S21}$$

where the Gini impurity g_i is defined by Eq (S4). Thus smaller values of g_i imply higher variance $\mathbb{V}\left[y_b^{(\Pi)}\right]$ of the window-smoothed scores obtained under random permutation Π (with mean unchanged). This increased variance will lead to the requirement of larger cutoff values θ for reporting motifs discovered in the unpermuted data with a given degree of confidence unless positions i with $g_i < g_{\min}$ are filtered out as described in Section S2.2.

S2.6 Permutation testing for multiple parameter combinations supplementary

Multiple combinations $\left(\kappa^{(\alpha)}, \lambda^{(\alpha)}, g_{\min}^{(\alpha)}\right)$ of the values of SArKS parameters may be explored (with α here indexing the set of desired combinations).

For any permutation π, let

$$\hat{y}_{\max}^{(\alpha,\pi)} = \max_{i \in I^{(\alpha,\pi)}} \left\{ \hat{y}_i^{(\alpha,\pi)} \right\} \tag{S22}$$

$$\hat{\hat{y}}_{\max}^{(\alpha,\pi)} = \max_{i \in I_{\text{spatial}}^{(\alpha,\pi)}} \left\{ \hat{\hat{y}}_{s_i}^{(\alpha,\pi)} \right\} \tag{S23}$$

(i.e., $\hat{y}_{\max}^{(\alpha,\pi)}$ is the highest filtered sequence-smoothed score obtained after permuting by π, while $\hat{\hat{y}}_{\max}^{(\alpha,\pi)}$ is similarly the highest filtered spatially-smoothed score). Then we suggest a simple method for setting thresholds $\theta^{(\alpha)}$ and $\theta_{\text{spatial}}^{(\alpha)}$ based on a set of randomly generated permutations $\{\pi_r\}$:

$$\theta^{(\alpha)} = \operatorname*{mean}_r \left\{ \hat{y}_{\max}^{(\alpha,\pi_r)} \right\} + z \operatorname*{stdev}_r \left\{ \hat{y}_{\max}^{(\alpha,\pi_r)} \right\} \tag{S24}$$

$$\theta_{\text{spatial}}^{(\alpha)} = \operatorname*{mean}_r \left\{ \hat{\hat{y}}_{\max}^{(\alpha,\pi_r)} \right\} + z \operatorname*{stdev}_r \left\{ \hat{\hat{y}}_{\max}^{(\alpha,\pi_r)} \right\} \tag{S25}$$

with higher values of z trading reduced sensitivity for lower false positive rates (in the examples analyzed in Section 3.2 we take $z = 4$). For the sake of simplicity we have generally used only one of these two thresholds for any particular combination of parameters α, setting $\theta^{(\alpha)} = -\infty$ if $\kappa^{(\alpha)} > 1$ or $\theta_{\text{spatial}}^{(\alpha)} = -\infty$ if $\kappa^{(\alpha)} \leq 1$ (i.e., if spatial smoothing is not employed).

In order to characterize the false positive rate associated with the entire set of analyses across all of the parameter settings employed while controlling for multiple hypothesis testing, a family-wise error rate (FWER) ϵ resulting from these thresholds can then be estimated by generating an independent set of R' permutations $\{\pi'_r\}$ and counting the number of permutations π'_r for which a nonempty set of k-mer motifs is identified using any of the parameter sets $\left(\kappa^{(\alpha)}, \lambda^{(\alpha)}, g_{\min}^{(\alpha)}\right)$. That is, writing

$$e = \left| \left\{ r \mid \{\alpha \mid I^{(\alpha,\pi'_r)} \cup I_{\text{spatial}}^{(\alpha,\pi'_r)} \neq \emptyset\} \neq \emptyset \right\} \right| \tag{S26}$$

(where $I^{(\alpha,\pi'_r)}$ and $I_{\text{spatial}}^{(\alpha,\pi'_r)}$ are defined respectively by Eq (S6) and Eq (S14) using the thresholds $\theta^{(\alpha)}$ and $\theta_{\text{spatial}}^{(\alpha)}$ determined using the original permutation set $\{\pi_r\}$) we can infer confidence intervals by noting that the random variable E instantiated in e satisfies $E \sim \text{Binom}(R', \epsilon)$ under the permutation test null hypothesis. We can thus derive confidence intervals (CIs) for the FWER (in the weak sense, as the permutation test represents a complete null hypothesis with no true positives [Farcomeni, 2008]) by applying the Clopper-Pearson method for estimation of binomial CIs.

S2.7 RNA-seq expression analysis supplementary

In order to test SArKS, we selected the *Mus musculus* neocortical neuron RNA-seq data set GSE63137 [Mo *et al.*, 2015] from Gene Expression Omnibus database [Barrett *et al.*, 2013] (https://www.ncbi.nlm.nih.gov/geo/). This data set contains detailed transcriptomic and epigenetic information from three functionally and neurochemically distinct classes of pooled neocortical neurons: principal excitatory neurons, parvalbumin (PV)-positive GABAergic neurons, and vasoactive intestinal peptide (VIP)-positive GABAergic neurons.

Because the position of the first exon can help pinpoint the TSS—and hence the DNA region containing the putative promoter—we reanalyzed the GSE63137 RNA-seq data using `kallisto` [Bray *et al.*, 2015] to quantify and normalize transcript level expression against Ensembl mouse cDNA reference GRCm38. Transcript species were filtered by mean expression to focus on those for

which reliable expression estimates could be made, retaining only transcripts for which at least 100 pseudocounts were obtained when summed across all samples and whose mean normalized expression met or exceeded the median of the transcript mean normalized expression levels. We also filtered out those transcripts which showed low variance across the full sample set, retaining only those for which the variance of normalized expression values met or exceeded the median such value across all transcript species [Bourgon *et al.*, 2010]. In order to simplify downstream analysis, only the isoform with highest mean expression level across all samples was retained for each detected gene. Finally, as previously analyzed epigenetic information on chromatin accessibility was available from the same study [Mo *et al.*, 2015], only transcripts for which the transcription start sites were located within ATAC-seq peaks (i.e., were accessible) for all examined neuron classes were retained for analysis. Imposing this condition minimizes the likelihood that epigenetic factors, rather than regulatory sequence characteristics, underlie the variations in gene expression between cell classes.

Differential gene expression was then assessed on normalized expression values via standard Student's t-test comparing data for PV neurons to data for excitatory and VIP neurons, with the resulting t-statistic providing a rough estimate of the gene's enrichment in PV neurons (score y_b for transcript b). One potential issue with the use of such t-statistics with small sample numbers—here, two samples associated with each neuron type—is that especially low within-group standard deviation estimates can result in very large magnitude t-statistics for a few genes. For example, the average estimated within-group standard deviation of the 76 genes (out of 6,326 surviving the filters described above; see Table 2) for which $|t_b| > 10$ (with $|t_b|$ ranging up to a maximum value of 49.6) was less than 30% of the average within-group standard deviation of the full set of 6,326 analyzed genes. Every one of the 76 genes with such high magnitude t-statistics had a within-group standard deviation estimate below the median value for the full gene set.

This phenomenon has previously led to the application of empirical Bayes methods [Smyth, 2004] using moderated t-statistics in place of standard t-statistics for calculating differential expression p-values. As we are here instead interested in using the t-statistics to derive word scores y_b, for which no particular distributional assumptions are required, we have adopted a simpler approach to prevent the few very large magnitude t-statistics from unduly influencing motif discovery by applying a ceiling of 10 on the magnitude of y_b:

$$y_b = \begin{cases} -10 & \text{if } t_b \leq -10 \\ t_b & \text{if } -10 < t_b < 10 \\ 10 & \text{if } t_b \geq 10 \end{cases} \tag{S27}$$

S2.8 Specifications used for running existing motif discovery algorithms

```
## ------------
## DREME options:
dreme \
 -p $pos_seq_fasta \
 -n $neg_seq_fasta \
 -oc $out \
 -png
## ------------

## ------------
## FIRE options:
perl fire.pl \
 --expfiles=$scores \
 --exptype=continuous \
 --fastafile_dna=$seq_fasta \
 --seqlen_dna=$seq_len \
 --nodups=1
## ------------

## ------------
## HOMER options:
homer2 denovo \
 -i $pos_seq_fasta \
 -b $neg_seq_fasta
## ------------

## ------------
## MOTIFREGRESSOR options:
MotifRegressor.pl \
 $scores \
 $seq_fasta \
 null 1 1 2 1 0 50 250 50 250 5 15 50 30 \
 $out
## _interpretation of MOTIFREGRESSOR parameters above_
##  null : background sequence distribution to be computed based on input sequences
##  1 : use column 1 from $scores to rank sequences
##  1 : use column 1 from $scores to perform regression
##  2 : data does need to be further log-transformed
##  1 : look for motifs in high-scoring (as opposed to low-scoring) sequences
##  0 : select fixed count of top motifs (as opposed to setting fixed score threshold)
##  50* : number of initial top motifs
##  250* : number of sequences with high values for confirmation
##  50* : (ignored since we are only interested in high-scoring motifs)
##  250* : (ignored since we are only interested in high-scoring motifs)
##  5* : minimum motif width
##  15* : maximum motif width
##  50* : number of seed candidate motifs
##  30* : number of motifs reported before regression
## _all parameters marked with * were set at the example (e.g.) values given in
##  MOTIFREGRESSOR's README.MR file_
## ------------
```

Table S1: **Unpermuted scores consistently exceed permuted scores only when motif is present.**

$\hat{y}_{\max} - \hat{y}^{(\pi)}_{\max}$	motif (+)	motif (-)
$-2/9$	0	7
$-1/9$	0	275
0	29	479
$1/9$	551	229
$2/9$	420	10

Distribution of simulated differences $\hat{y}_{\max} - \hat{y}^{(\pi)}$ obtained by suffix array kernel smoothing when CATACTGAGA motif is embedded into 10 high score sequences (motif (+) column) or when it is not (motif (-) column). The values $\hat{y}_{\max}$ were calculated by smoothing the unpermuted sequence scores y_b, while the values $\hat{y}^{(\pi)}_{\max}$ were obtained using permuted sequence scores $y_{\pi(b)}$. When motif is included, $\hat{y}_{\max} - \hat{y}^{(\pi)}$ tends to be positive—i.e., unpermuted smoothed scores usually exceed permuted—while when motifs are not present the distribution is symmetric about 0, reflecting the lack of signal for SArKS to detect.

S3 Results and Discussion supplementary

S3.1 Illustration using simulated data supplementary

To demonstrate that these results are not a quirk of a single simulation, we repeated the process of generating 30 random sequences, embedding the motif CATACTGAGA into the last 10 of them, and then applying suffix array kernel smoothing to the sequence scores 1000 times. In 971 iterations, the maximum value

$$\hat{y}_{\max} = \max\{\hat{y}_i \mid g_i \geq g_{\min}\} \tag{S28}$$

calculated using the unpermuted sequence scores exceeded the maximum value

$$\hat{y}^{(\pi)}_{\max} = \max\{\hat{y}^{(\pi)}_i \mid g_i \geq g_{\min}\} \tag{S29}$$

obtained using one set of randomly permuted sequence scores per iteration. The full distribution of the differences $\hat{y}_{\max} - \hat{y}^{(\pi)}_{\max}$ is shown in the motif (+) column of Table S1.

We also examined the results of SArKS applied to simulated data in which no motif was present to find; for this purpose, we repeated an amended version of the simulation process with the sole modification of omitting the motif embeddings again 1000 times. The distribution of $\hat{y}_{\max} - \hat{y}^{(\pi)}_{\max}$ for these no-motif simulations is presented in the motif (-) column of Table S1. In this case, $\hat{y}_{\max}$ exceeded $\hat{y}^{(\pi)}_{\max}$ in only 239 of the simulations, while $\hat{y}^{(\pi)}_{\max}$ exceeded $\hat{y}_{\max}$ in 282 simulations, with equality between the two holding in the remaining 479 iterations. The symmetry of the distribution of $\hat{y}_{\max} - \hat{y}^{(\pi)}_{\max}$ around 0 in the motif (-) case is to be expected since the scores y_b are independent of the sequences w_b whether permuted or not if no motifs are included. By contrast, the strong asymmetry of the distribution of $\hat{y}_{\max} - \hat{y}^{(\pi)}_{\max}$ when the motif is present demonstrates the ability of the permutation approach to differentiate a true signal from background noise.

S3.2 DNA motifs associated with neuron subtype-specific expression supplementary

S3.2.1 Benchmark comparisons to existing algorithms

An alternative approach to assessing motif discovery efforts is to compare the motifs identified algorithmically to databases of known TF-binding motifs, such as JASPAR [Mathelier *et al.*, 2015]. In the context of our analysis of the Mo 2015 RNA-seq data, we can also check whether or not

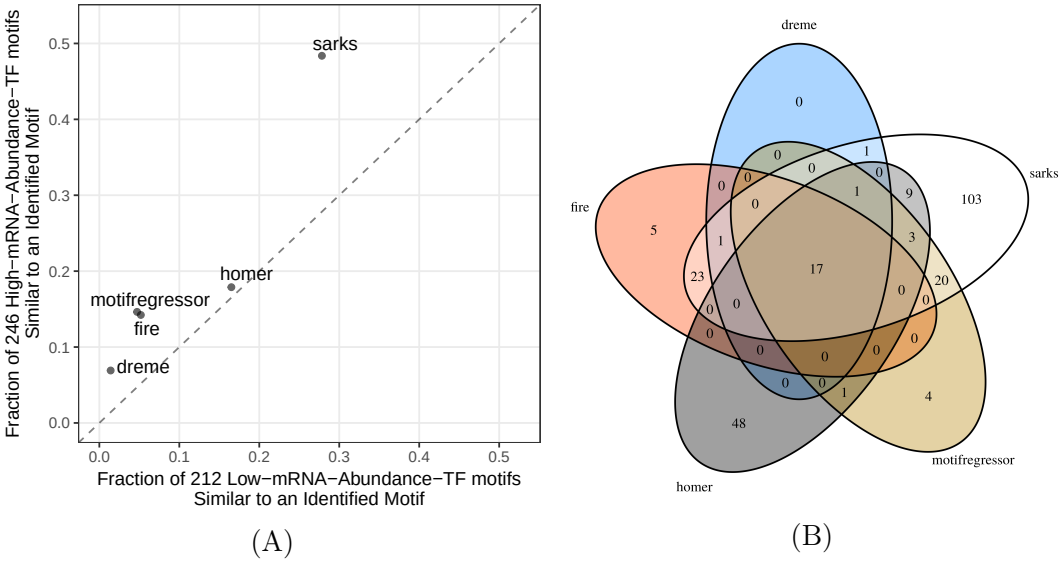


Figure S2: **Discovered motifs overlap with known transcription factor binding sites. (A)** Fractions of JASPAR-annotated TFs for which the algorithms indicated identified a motif with significant `tomtom` similarity to the corresponding JASPAR binding motif. Vertical axis: fractions of the 246 JASPAR-annotated TFs whose measured expression in parvalbumin neurons passed applied filters (see text); horizontal axis: fraction of the 212 JASPAR TFs with measured parvalbumin neuron expression below one of expression filters. **(B)** Venn diagram indicating the counts of JASPAR TF-binding motifs m for which all algorithms a in a particular subset $A \subset$ {dreme, fire, homer, motifregressor, sarks} identify a motif m'_a exhibiting significant `tomtom` similarity to m.

mRNA encoding TFs whose binding sites are similar to discovered motifs are present among the parvalbumin neuron transcripts. For this purpose, we employed a filter similar to that used in Section S2.7, requiring there to be at least one distinct mRNA transcript for the TF gene with at least 100 pseudocounts summed across the six neuron samples and for which the mean TPM-normalized estimated expression level in the parvalbumin samples $\geq$ the median value of such means for all Ensembl-annotated transcripts. Fig S2A shows the fraction of JASPAR TF-binding sites for which similar motifs were discovered by each algorithm for JASPAR TFs that passed the RNA-seq expression filter on the vertical axis. The horizontal axis of Fig S2A indicates instead the fraction of JASPAR TFs with Mo 2015 mRNA levels below the filter requirements (motif similarity was determined using `tomtom` [Gupta *et al.*, 2007] with a q-value cutoff of 0.1). Fig S2A is similar to a receiver-operating characteristic plot in which the motif discovery algorithms are regarded as classifying JASPAR motifs as positive when they show sufficient similarity to any of the discovered motifs; the distance of a point above the diagonal thus indicates the degree to which an algorithm preferentially identifies binding motifs for TFs showing high RNA-seq expression levels. SArKS identifies motifs similar to a larger fraction of JASPAR than do the other algorithms while maintaining a preference for motifs for highly expressed TFs.

Fig S2B shows the degrees of overlap in the sets of JASPAR motifs with similarities among the motifs identified by the motif discovery algorithms. HOMER identified motifs similar to the bulk—49 out of 58 total—of the JASPAR motifs for which SArKS did not find any similar pattern but at least one of the remaining algorithms did. Of these 49 JASPAR motifs, 48 were not identified as similar to any motifs discovered by any of the algorithms other than HOMER.

S4 Future Directions

S4.1 Gapped motif detection

While lexical sorting of suffixes assembles occurrences of the same k-mer into a block of adjacent index positions i, gapped motifs such as

$$u = u_0 * u_{\text{gap}} * u_1 \tag{S30}$$

in which there is significant variability in the characters appearing within the internal substring u_{gap} will be scattered into distinct subblocks dispersed within the larger superblock corresponding to their common prefix u_0. This dispersion can dilute the apparent correlation $\hat{y}_i$ between motif and score by mixing non-matching suffixes in with those corresponding to u within the range of the smoothing kernel.

While the technique described in Section S2.4 ameliorates this problem, it does not specifically focus on the important situation where a head motif u_0 is always followed by the same tail motif u_1 after the variable region u_{gap}. Such gapped motifs might be discovered using SArKS by first applying a relatively relaxed threshold θ (which may on its own admit many false positives) and then examining the tail sequences $u_{\text{gap}} * u_1 * \cdots$ following it for evidence of an enriched sequence u_1, removing candidate head sequences for which no such corresponding tails can be found. In this way, the ability of SArKS to detect motifs with particularly variable internal positions may be improved.

S4.2 Synergistic motif detection

Because the regulation of eukaryotic gene expression may involve interactions among multiple motifs [Wunderlich and Mirny, 2009], it is of interest to discover motifs that work together synergistically to promote cell-type-specific gene expression. To achieve this objective, we can extend the kernel smoothing methods introduced here by, for example, utilizing multiple index kernel regression models such as

$$\hat{y}_{ij} = \frac{\sum_{k,l} K_{ijkl}\, y_{b_k}}{\sum_{k,l} K_{ijkl}} \tag{S31}$$

where the 4-index kernel K_{ijkl} might be chosen to satisfy constraints along the lines of

$$K_{ijkl} = \begin{cases} 0 & \text{if } (|i-k| > \kappa_2) \vee (|j-l| > \kappa_2) \\ 0 & \text{if } (b_i \neq b_j) \vee (b_k \neq b_l) \\ 1 & \text{otherwise} \end{cases} \tag{S32}$$

That is, i and k must correspond to suffixes with sufficiently similar prefixes (as must j and l), while i and j must come from the same word (as must k and l). Fig S3 depicts four suffixes with indices i, j, k, l which would contribute to the smoothed score $\hat{y}_{ij}$. Intuitively, if the beginning of the suffixes starting at s_i and s_j were two motifs which work together synergistically—and thereby lead to elevated scores y_{b_i} for the sequence $b_i = b_j$ which they are found in—we might expect to see elevated scores y_{b_k} for sequences $b_k = b_l$ containing the similar motifs prefixing the suffixes starting at s_k and s_l.

S4.3 Improved selection of input sequences

While we have investigated sequences near the TSS, it is well known that regulatory elements may lie quite far from the genes they affect [Sanyal *et al.*, 2012]. It would be advantageous to develop more sophisticated approaches for (1) the identification of genomic regions most likely to contain regulatory elements and (2) the linking of potential regulatory element-dense-regions to genes whose expression they govern. Such approaches would likely benefit from consideration of information on

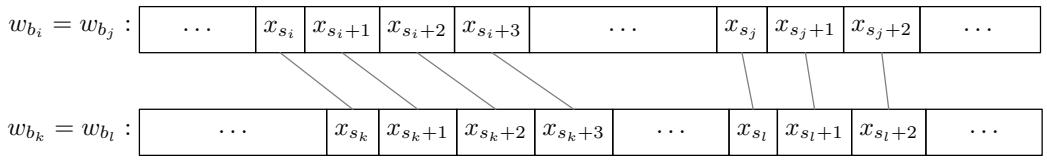


Figure S3: **Synergistic motif visualization.** Visualization of a quartet (i, j, k, l) satisfying constraints (S32): the suffix $x[s_i, |x|)$ originating from sequence w_{b_i} shares a common prefix, or motif, with the suffix $x[s_k, |x|)$ originating from sequence w_{b_k}; a second pair of suffixes $x[s_j, |x|)$ and $x[s_l, |x|)$ sharing a (different) motif in common are also found in the sequences $w_{b_j} = w_{b_i}$ and $w_{b_l} = w_{b_k}$, respectively. Each box contains a single character (or base, for DNA sequences), with motifs indicated by connecting two series of boxes composing matching substrings in different w_{b_i}.

evolutionary conservation [Xie *et al.*, 2005] and epigenetic modification [Long *et al.*, 2013] near genes of interest.

S4.4 Other applications of SArKS

While we have tested SArKS as a method for identifying candidate cell-type specific regulatory motifs, it could also be applied to sequence motifs associated with state dependent changes in activated neurons of a single class as well as to differential gene expression in cancer and in specimens that have been exposed to varying physical or chemical stimuli. We also anticipate uses far afield from analysis of biological sequences, including motif discovery in time series data [Fu, 2011], or, by considering node or edge sequences produced by random walks, analysis of complex network structure [Masoudi-Nejad *et al.*, 2012].