S2 Methods

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 j^{th} 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
ω_i	0-based position of suffix with suffix array index i within block b_i
$\hat{y}_i$	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 Limiting the impact of intra-sequence repeats

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 where 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 Equation (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:

$$1 - g_{\min} = (1 + \gamma) \left(1 - \underset{i}{\text{median}}\, g_i\right) \tag{S5}$$

As shown in Equation (S21), setting $g_{\min}$ to satisfy Equation (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 to reduce redundancy in reported motif set.** 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 Reducing redundancy in reported motif set

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 Figure 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 shorter k-mers nested inside longer peak motifs

Cases in which both k-mer $x[s_i, s_i+k)$ (e.g., CATACTGAGA in Figure S1) and its sub-$(k-m_1-m_2)$-mer $x[s_i+m_1, s_i+k-m_2)$ (with $m_1 > 0, m_2 \geq 0$; e.g., TACTGAGA in Figure S1 with $m_1 = 2, m_2 = 0$) are individually identified can be resolved to report only the longer k-mer by removing any index $i \in I$ (defined by Equation (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 to match longer motifs from distinct peaks

Besides two cases of nested k-mers which may be removed from the reported motif set by the method of Section S2.3.1 (ATACTGAGA and TACTGAGA), Figure S1 also depicts a shorter k-mer ATACTG derived from a distinct occurrence of the same longer k-mer (CATACTGAGA). 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. Equation (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 Figure 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 to identify multi-motif domains

SArKS identifies 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

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

(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 Equation (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 Equation (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:

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

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

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

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

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

Equation (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 $\hat{y}_i \ge \theta_{\text{spatial}}$, 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. Equation (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

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

S2.5 Permutation testing to establish significance of motif set

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 Equation (4) (and also $\hat{\hat{y}}_{s_i}^{(\pi_r)}$ using Equation (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 Equation (S4) in Section S2.2. 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 Equation (S3) and for all b

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

Equation (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 Equation (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 to set thresholds for multiple parameter combinations

Multiple combinations $\left(\kappa^{(\alpha)}, \lambda^{(\alpha)}, g_{\min}^{(\alpha)}\right)$ of the values of SArKS parameters may be explored (with α indexing the set of desired combinations); for example, Sections 3.2.1-3.2.2 discuss the parameters used in the benchmarking examples herein and the rationales for their selection.

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 3.2.1 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 Equation (S6) and Equation (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

S2.7.1 Assigning PV differential expression scores for Mo 2015 data set

In order to test SArKS, we selected the *M. musculus* neocortical neuron RNA-seq data set GSE63137 (Mo *et al.*, 2015) from Gene Expression Omnibus (GEO) 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-positive (PV) GABAergic neurons, and vasoactive intestinal peptide-positive (VIP) 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`

	Mo 2015	Close 2017
All	111,669	23,045
Detected	73,912	15,490
+ Highly Expressed	37,721	6,939
+ Highly Varying	29,164	
- Duplicate Isoforms	11,857	
+ Accessible	6,326	
Final SArKS Set	6,326	6,939

Table S1: **Filters applied to select gene sets for SArKS analysis.** The Mo 2015 data set (bulk RNA-seq) was realigned and analyzed at isoform level, hence counts in first column indicate distinct transcripts or isoforms. For the single-cell RNA-seq Close 2017 data set, the original gene-level alignment counts were analyzed; counts in second column indicate distinct genes. No variance filter was applied for the Close 2017 data set, as none of the 6,939 highly expressed genes exhibited low estimated variance. Epigenetic accessibility data was available for the Mo 2015 samples but not the Close 2017 samples.

(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 transcripts that showed low variance across the full sample set, retaining only those for which the estimated variance $\hat{\sigma}_b^2$ of normalized expression values met or exceeded median$\{\hat{\sigma}_b^2\}$ 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 per gene. Finally, as based on chromatin accessibility data (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 analyzed. This accessibility-based filter reduced the likelihood that epigenetic features, rather than regulatory sequences, determine the variations in gene expression between cell classes.

Differential gene expression was assessed using 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 an estimate of a 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 with $|t_b| > 10$ (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 (Table S1). 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.

The phenomenon of low within-group variance estimates leading to inflated test statistics 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:

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

S2.7.2 Assigning DCX differential expression scores for Close 2017 data set

We also examined an RNA-seq data set comparing transcriptomes of *in vitro*-induced human embryonic stem cells and the resulting cultured interneurons (Close *et al.*, 2017). We applied SArKS to identify promoter motifs associated with elevated gene expression in doublecortin-positive (DCX+) interneurons. We restricted our analysis to the post-induction day 54 (D54) timepoint, where most of the DCX+ neurons were post-mitotic and GABAergic, and for which the largest total number of cells had been profiled, minimizing the within-group expression variations.

We used the normalized gene expression levels from GEO (Barrett *et al.*, 2013) records for this data set (accession GSE93593). We chose not to reanalyze the sequencing data in this case because we did not want to split the read counts per cell—which, given the large numbers of cells observed, tend to be much lower than read counts per sample in bulk RNA-seq—across multiple distinct transcripts for each gene. We found 15,490 genes for which (1) nonzero aligned read counts were detected in at least one (out of 585) analyzed cells and (2) a unique entry was found in the GRCh38 annotation of the human genome. We retained the 6,939 genes from this set for which the average aligned read count per cell was ≥ 25 for further analysis (Table S1). As the variance of the log2-transformed transcripts-per-million (TPM) normalized expression levels was quite high (≥ 1 for 6,852 of the 6,939 genes, ≥ 0.5 for all 6,939 genes), we did not apply any variance filter for this data set. As no epigenetic information was available, no accessibility filtering could be conducted.

For the filtered high-read-count gene set, differential expression was assessed via a simple two group t-test comparing the DCX+ cells to the DCX- cells and SArKS scores were assigned according to Equation (S27), just as was done for the Mo 2015 data set.

S2.8 Specifications for running existing motif discovery algorithms

Existing algorithms were run at their default parameter settings (defined either within the source code or in associated documentation), with two exceptions: (1) 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 anticorrelated motifs by default). (2) STEME was run in discriminative mode using a high order Markov model on the negative sequences exactly as suggested in the online documentation (https://pythonhosted.org/STEME/using.html); however, STEME's implementation requires pre-specification of the number of motifs to report, defaulting to a single motif if unspecified. Given that (Reid and Wernisch (2014)) extensively compared STEME to DREME with the finding that the two were generally comparable in performance, we took the upper bound on the observed DREME motif set size (10 motifs) as the number of motifs for STEME to report.

```
## ------------
## 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 not 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_

## ------------
## STEME options:
steme \
 --output-dir=$out \
 --num-motifs=10 \
 --bg-model-order=5 \
 --bg-fastafile=$neg_seq_fasta \
 $pos_seq_fasta
## _see https://pythonhosted.org/STEME/using.html_
```

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

Table S2: **Unpermuted scores consistently exceed permuted scores only when motif is present.** 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

S3.1 Illustration of SArKS using simulated data

To demonstrate that the results of Section 3.1 are not a quirk of a single simulation, we repeated the process of (1) generating 30 random sequences, embedding the motif CATACTGAGA into the last 10 sequences, and (2) applying SArKS to the sequences and 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 S2.

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 1000 times, omitting the motif embeddings. The distribution of $\hat{y}_{\max} - \hat{y}^{(\pi)}_{\max}$ for these no-motif simulations is presented in the motif (-) column of Table S2. 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.

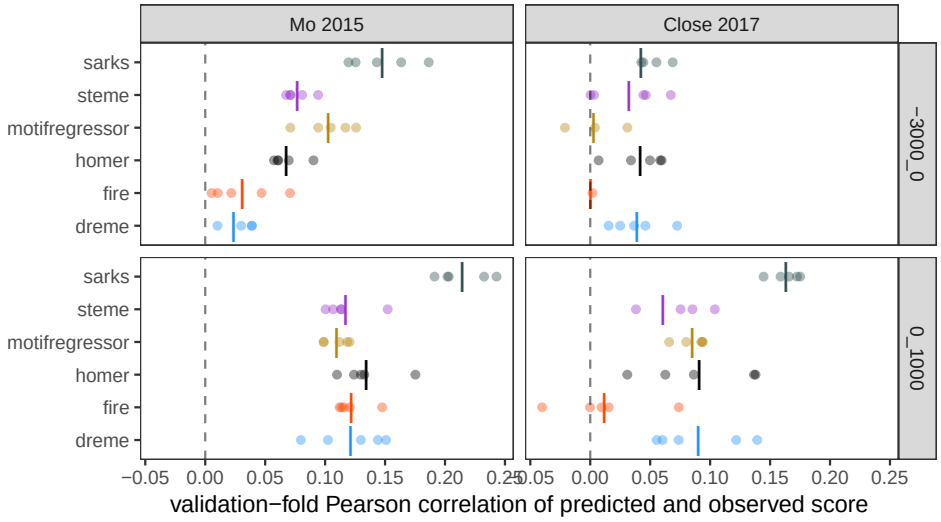


Figure S2: **Motif regression model predictions correlate with gene specificity scores in held-out validation subsamples.** Each of five cross-validation folds is plotted as separate point for each algorithm; fewer than five points are shown for corresponding algorithm if it failed to identify any motifs in one or more cross-validation folds. Upper panels: regions 3kb upstream of TSS; lower panels: regions 1kb downstream of TSS. Vertical lines indicate mean Pearson correlation across all folds (including a value of 0 for any fold in which algorithm failed to identify any motifs).

S3.2 Uncovering promoter motifs associated with differential gene expression

S3.2.1 Benchmark comparisons to existing algorithms

The cross-validated regression modeling strategy described in Section 3.2.3 builds a single regression model based on the concatenated upstream and downstream motif count vectors. We also built two more separate regression models—one using as feature set only the upstream motif counts, the other only the downstream motif counts—for each of the two data sets, obtaining the results presented in Figure S2. SArKS generally outperformed the other algorithms in these comparisons, though for the upstream analysis of the Close 2017 data, DREME and HOMER offer similar performance; all of the algorithms have their poorest performance in this particular analysis. For both data sets the regression model predictions on the held-out validation folds are noticeably better in the downstream analyses than the upstream analyses, as discussed in Section 3.2.3.

We used `tomtom` (Gupta *et al.*, 2007) to compare the pooled motif sets identified by each algorithm and detected overlap between motifs sets by algorithm (S3). There exists a significantly similar ($q \leq 0.1$) SArKS-identified motif for the majority of motifs identified by any of the existing algorithms in the Mo 2015 data set. For the Close 2017 data set, at least 50% of the motifs identified by DREME, FIRE, MOTIF REGRESSOR, and STEME can be paired with a significantly similar SArKS motif, though this is true for only 39% of HOMER-identified motifs.

An alternative benchmarking approach is to compare the motifs identified algorithmically to databases of known TF-binding motifs, such as JASPAR (Mathelier *et al.*, 2015). For the presence of a TF-binding site to be biologically relevant in a cell, it is necessary for the

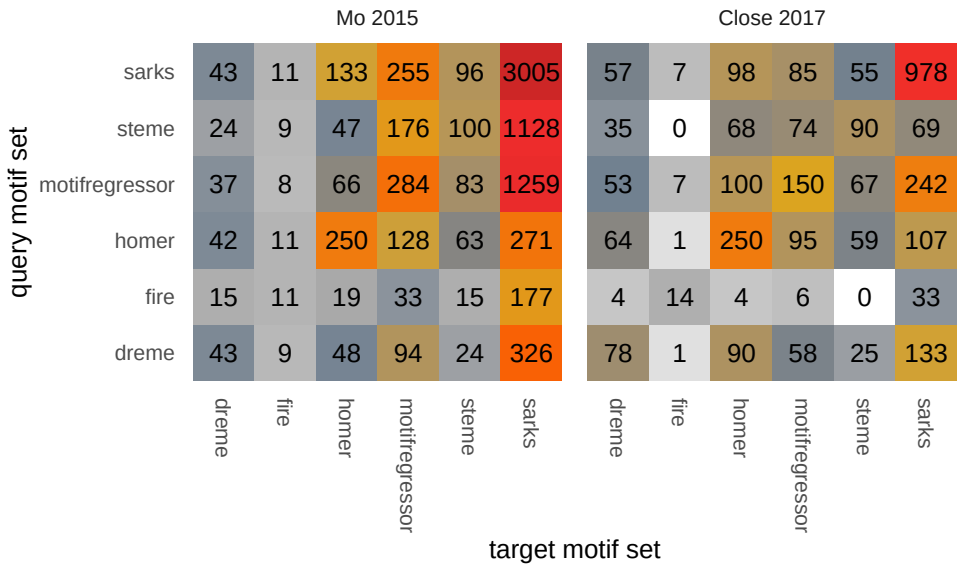
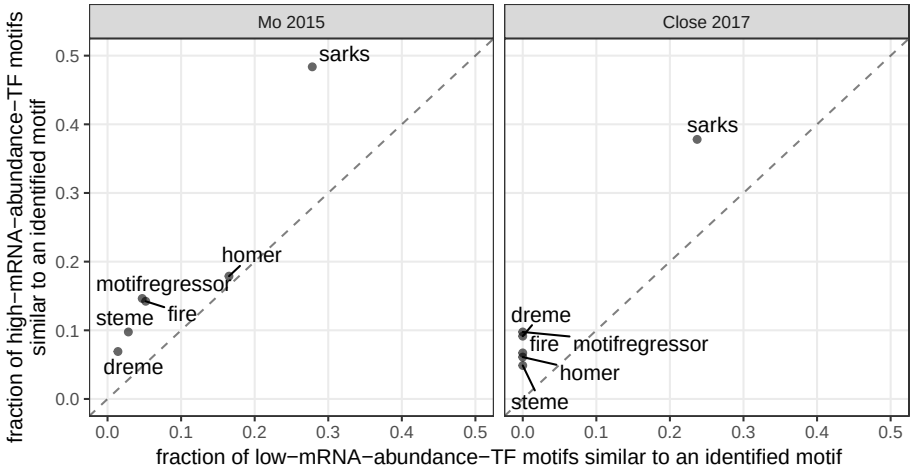


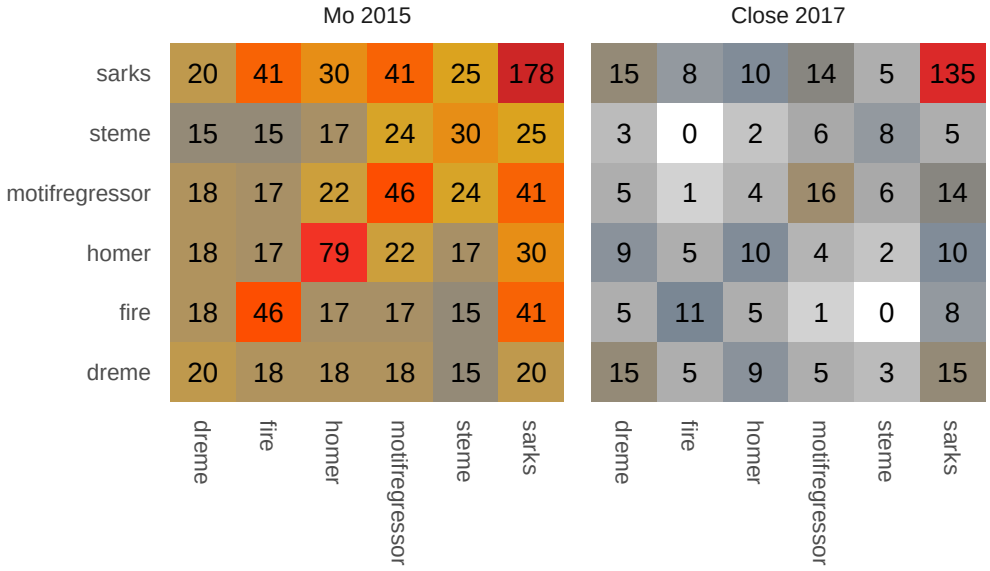
Figure S3: **Counting motif similarities shows substantial overlap between algorithms.** Each cell indicates the count of motifs identified by the target motif set algorithm for which there is a motif in the query set with significant `tomtom` similarity ($q \leq 0.1$). Cells are colored according to the numbers they contain.

TF itself to be present as well. In the context of our analysis of the two RNA-seq data sets, we checked whether or not mRNA encoding TFs whose binding sites were similar to discovered motifs are enriched among either the PV neuron (Mo 2015) or DCX+ cell (Close 2017) transcripts. We classified a TF gene as enriched if there was at least one distinct mRNA transcript for the gene with (1) at least 100 reads (or pseudocounts for the Mo 2015 set) and (2) for which the mean TPM-normalized estimated expression level in either the PV samples (Mo 2015) or DCX+ cells (Close 2017) $\geq$ the median of the genewise means for all measured transcripts/genes in the relevant data set. Figure S4A 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 indicates the degree to which an algorithm preferentially identifies binding motifs for TFs showing high RNA-seq expression levels in the target cell population. 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.

Figure S4B illustrates the overlaps between the sets of JASPAR motifs with similarities among the motifs identified by the motif discovery algorithms: For all algorithms applied to the Close 2017 data set and all but HOMER in the Mo 2015 data set, the set of JASPAR motifs with significant similarity ($q \leq 0.1$) to one of the algorithm-identified motifs overlaps by more than 50% with the set of JASPAR motifs significantly similar to a SArKS motif. The degree of overlap between the JASPAR matches among the various algorithm motif sets tends to be higher than the degree of overlap directly between the motif sets themselves. This suggests that the presence of a similar JASPAR motif may provide supporting evidence that a given detected motif is a not a false positive.



(A)



(B)

Figure S4: **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 ($q \leq 0.1$) to the corresponding JASPAR binding motif. Vertical axis: fractions calculated using only the JASPAR-annotated TFs whose measured expression in either PV neurons (left panel) or DCX+ cells (right panel) were in top 50% by mean normalized expression (TPM) and had at least 100 associated reads. Horizontal axis: fractions calculated using only the remaining JASPAR-annotated TFs with measured expression below these expression filters. **(B)** Each cell indicates the count of JASPAR motifs for which there is a motif in both of the indicated algorithm motif sets with significant `tomtom` similarity ($q \leq 0.1$). Cells are colored according to the numbers they contain.

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%

Table S3: **Gini index filters remove small fractions of suffix array positions.** Fraction of suffix array positions i for which Gini impurity values $g_i \geq g_{\min}$, with $g_{\min}$ selected according to Equation (S5) (applied to Mo 2015 data set).

S3.2.2 Case study: analysis of SArKS results for Mo 2015 data set

The values of $g_{\min}$ obtained for the analysis of the Mo 2015 gene set (6,326 genes remaining after application of filters described in Section S2.7.1), along with the fraction of suffix array index values i for which $g_i \geq g_{\min}$, are listed in Table S3.

S3.2.2.1 Top motif identified in sequences downstream of TSS

The highest $\hat{y}_i$ value obtained—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 fact a large fraction of the motifs associated with identified peaks in $\hat{y}_i$ identified in the downstream analysis exhibit significant similarity to one of the JASPAR motifs ESRRB, ESRRA, or ESRRG, as is illustrated in Figure S5B. The ESRR(A/B/G) TFs are all members of the estrogen-related receptor family; there is evidence that these receptors are involved in brain functions including synaptic transmission, neuronal firing, and mitochondrial biogenesis (Saito and Cui, 2018). 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 (Figure 3A), as we noted that all of these algorithms identified motifs similar to each of these JASPAR motifs (Section 3.2.3).

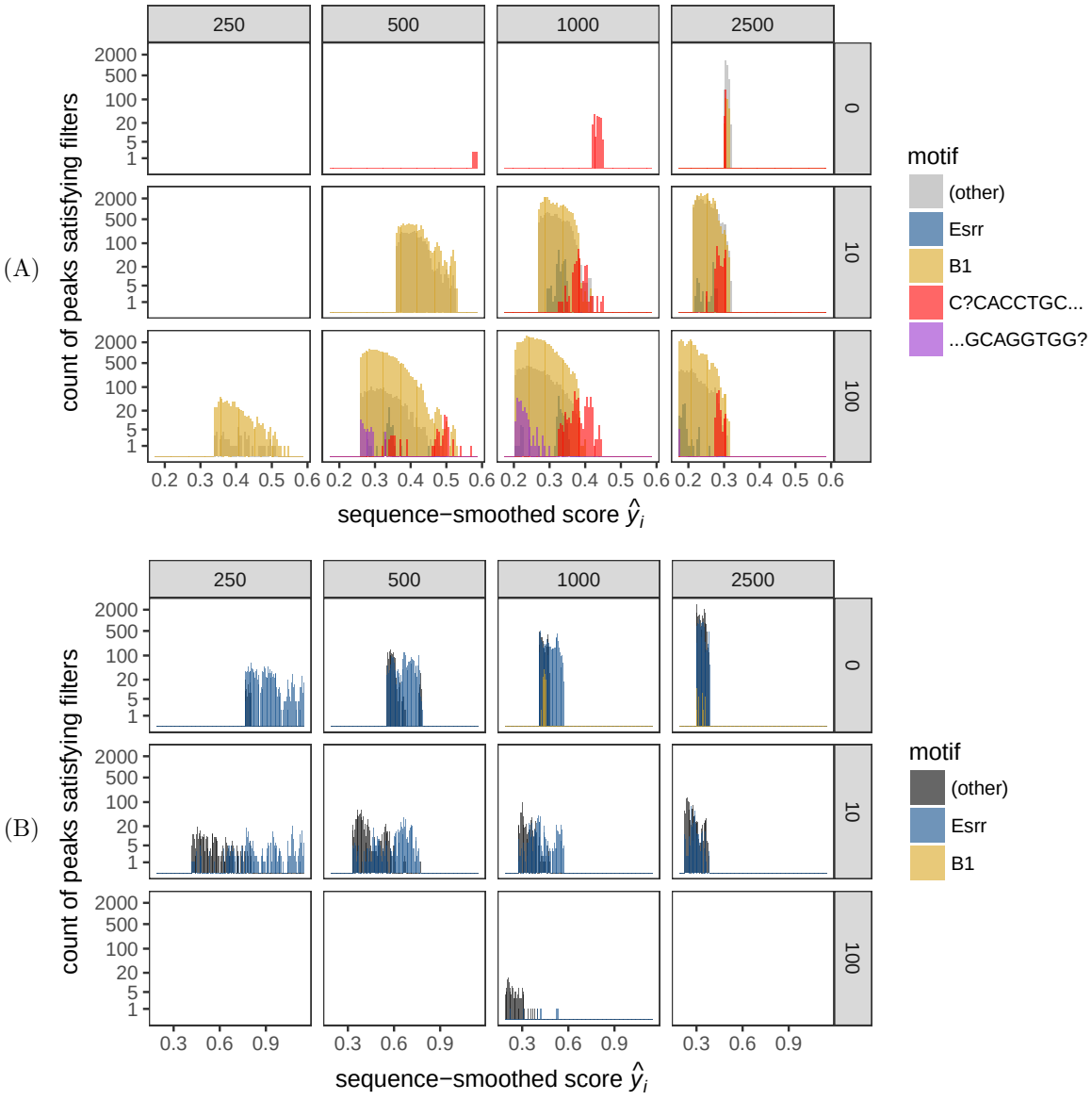


Figure S5: **Contributions of top motifs to peak composition. (A)** Log-scaled histograms of peaks $i \in I$ (or I_{spatial} when spatial smoothing is employed) identified in **upstream** analysis for which corresponding k-mer motifs: (1) are prefixed with CACCTGC or CCACCTGC (indicated in red) or are 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.

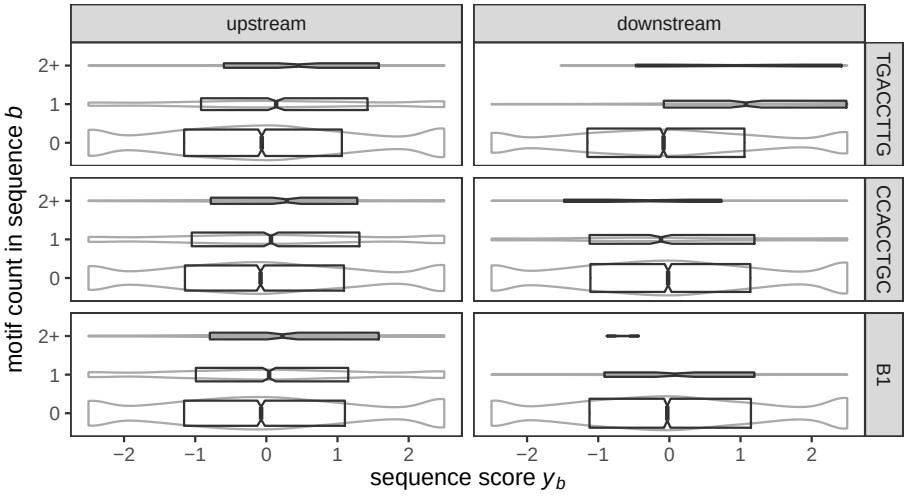


Figure S6: **PV-specific differential expression scores are higher in sequences containing one or more copies of top SArKS motifs** Each plot shows distribution of sequence scores split by the number of occurrences of motif indicated by the row label (TGACCTTG, CCACCTGC, or B1) found in the sequence range indicated by the column label (upstream or downstream). The first two motifs—k-mers TGACCTTG and CCACCTGC—were counted using regular expression matching (allowing matches on either forward or reverse strands), while B1 counts were assessed using `blastn` (percent identity $\geq 90\%$, alignment length ≥ 70). Distributions are summarized by notched boxplots (area scaled to square root of sequence count; notch width is 1.57 times the interquartile range (IQR) divided by square root of sequence count) laid over kernel density estimates drawn as gray violins (area scaled to sequence count). Scores ≤ -2.5 or ≥ 2.5 are plotted at -2.5 or 2.5, respectively, in order to clearly show the bulk of the distribution, which falls within this range. For the motif CCACCTGC derived from analysis of the upstream sequences, the scores for the downstream sequences containing the k-mer do not show the same upward shift in the score distribution as do those for the upstream sequences with one or more occurrence. On the other hand, the score distribution for both TGACCTTG-positive upstream sequences and TGACCTTG-positive downstream sequences is shifted upwards, though the shift is notably larger in the downstream sequences, explaining the prominence of this motif in the downstream analysis.

S3.2.2.2 Top motifs identified in sequences upstream of TSS

ESRRB/ESRRA/ESRRG binding motifs were also 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 (Figure S5A). Figure S6 sheds some light on this: the distribution of sequence scores y_b for downstream sequences containing one or more copies of the top SArKS octamer TGACCTTG is shifted upward to a much higher degree than is the the distribution of y_b values for upstream sequences containing TGACCTTG.

Instead, 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 (specifically, the E12-box variant CACCTG (Bouard *et al.*, 2016); we note that the significant SArKS peak set contains many peaks corresponding to the 7-mer CACCTGC as well as

the longer octamer adding the extra initial C). Comparison of CCACCTGC with known motifs from the JASPAR database using `tomtom` finds some evidence of similarity to 10 TF-binding motifs (SNAI2, MAX, SCRT2, SCRT1, TCF3, MNT, Id2, MAX::MYC, TCF4, and FIGLA; q-values of 0.14 for each), though no similarities significant at $q \leq 0.1$. Unlike the case for the ESRR(A/B/G) motifs discovered in the downstream analyses, for which all of the benchmarked algorithms detected a matching motif, only one of the other algorithms (HOMER) detected a motif similar to either CACCTGC or CCACCTGC (`tomtom` $q \leq 0.1$; no other algorithm produced any motifs matching even at $q \leq 0.5$).

S3.2.2.3 B1 SINE sequence identified through MMD analysis

As the octamer 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 (Figure S7A); three of the five 100-mers thus aligned were very similar (Levenshtein distance ≤ 7) to the 99-mer consensus sequence constructed. Furthermore, the consensus sequence also contains CCTGGAAC and CCAGGCTG (reverse complement of CAGCCTGG).

A `blast` 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 appearing 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 further suggested that SINEs function as transcriptional enhancers (Ichiyanagi, 2013; Elbarbary *et al.*, 2016; Ge, 2017).

Figure S5A indicates the number of SArKS-identified peaks that fall within `blast` hits between the upstream sequences w_b and the B1 SINE consensus sequence as well as the numbers of peaks corresponding to the top motifs discussed above. The upstream SArKS peaks derived from analyses involving spatial-smoothing ($\lambda \in \{10, 100\}$) are dominated by B1 sequences, many including the CCACCTGC motif or its reverse-complement.

S3.2.2.4 SArKS motifs correspond to variations on B1 sequence

Figure S7B 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 and by whether they are matched or mismatched to the B1 consensus at each position. The k-mer CCACCTGC itself is not quite a perfect match to the canonical 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 Figure S7B—one of the few positions at which there are more mismatches than matches. This G to A substitution creates the above noted E-box sequence CANNTG, while the unmodified octamer CCGCCTGC does not match any JASPAR motifs at $q \leq 0.5$. This highlights the ability of SArKS to discover potentially functionally significant variations within a recurring 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 several JASPAR motifs (TBX21, EOMES, TBX15, TBX1, and TBX2), while the unmodified B1 nonamer

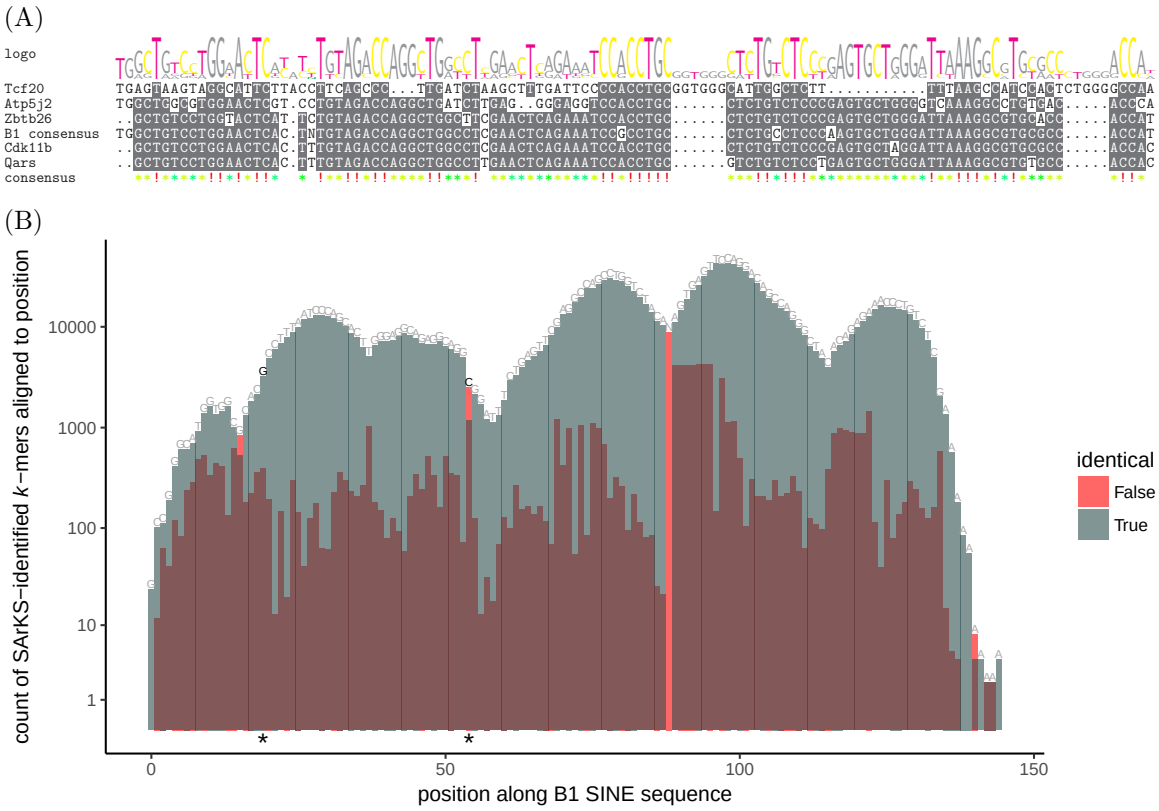


Figure S7: **SArKS-discovered motifs within B1 SINE elements. (A)** Multiple sequence alignment (`muscle`) of 100-mers surrounding top CCACCTGC motif peaks with reverse-complement of B1 consensus sequence. Associated genes are indicated to the left. Gray highlighting: $\geq 50\%$ agreement in the multiple 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 against 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. 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.

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

Table S4: **SArKS-based regression modeling assists in selecting candidate upstream regions for promoting PV-specific expression.** 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.

GCACGCCTT again shows no similarity to any JASPAR motifs at $q \leq 0.5$, again suggestive that specific SINE variants may promote differential gene expression.

S3.2.2.5 SArKS-based candidate promoter selection

Finally, 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 3.2.3 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 S4 shows a sequence of filters in which these regression scores were applied alongside other relevant criteria to select candidate PV-specific promoter regions. The promoter regions associated with the genes ATP5SL, GPRC5B, IFT27, KCNH2, MAFB, PAQR4, SLC29A2, SYT2, TBC1D2B, TMEM186, and TTC39A comprise the 11 candidates (from the final row of Table S4) selected for further experimental validation. Table S5 shows which of the top motifs discussed above are present in each of the candidate promoter regions: all regions except those for GPRC5B and MAFB contain at least one match for the ESRRB motif, while several also contain one or more copies of the E-box sequence and/or a match to the B1 SINE sequence. The promoter for IFT27 contains a match to a variant the B1 sequence with the substitution creating the E-box sequence CACCTGC. It is worth noting that there are many other SArKS motifs contributing to the promoter ranking model used here. Indeed, in accord with the principle that there is likely to be more than one way for combinations of motifs to achieve expression specificity, the candidate promoters for the genes GPRC5B and MAFB are ranked highly based exclusively on motifs other than the highest scoring ones.

Promoter	ESRRB	C?CACCTGC (E-box)	B1
ATP5SL	3	1/3	0
GPRC5B	0	0/1	0
IFT27	1	0/2	1
KCNH2	2	0	0
MAFB	0	0	0
PAQR4	2	2	0
SLC29A2	1	0	1
SYT2	2	1	0
TBC1D2B	3	1	0
TMEM186	2	0	1
TTC39A	2	0/1	0

Table S5: **Selected candidate promoter regions contain different combinations of top motifs.** Counts for the JASPAR motif ESRRB—the best JASPAR match to the top SArKS motif TGACCTTG—were assessed using `fimo`, while counts of the E-box sequence CCACCTGC or its reverse complement were assessed using simple string matching. If a promoter had additional matches to the substring CACCTGC (on either strand) omitting the initial C, a second count for this reduced match is indicated after a forward slash. Matches for the B1 SINE sequence were counted using `blast` requiring a minimum 90% sequence identity and 70 bp alignment length.

S3.3 Computational complexity and scalability of SArKS

One of the major motivations behind SArKS' method of discovering motifs—searching for blocks of lexicographically similar suffixes derived predominantly from high-scoring sequences—lies in the scalability of suffix-based methods. The number of suffixes of a string (or set of strings) scales linearly with the length of the string(s) involved: as a result, the steps involved in the SArKS algorithm for identifying significant peaks scale linearly in both runtime and memory space with the combined size of the set of input sequences. We discuss this in more detail below. We then discuss the complexity of the later steps involved in extracting information regarding specific motif k-mers from the significant SArKS peak set.

The existing implementation of SArKS generates and then stores in memory the full suffix array of the concatenated sequence $x = w_0 * \ldots w_{n-1}$: this step is asymptotically linear in the length of the concatenated sequence both in terms of runtime and memory (Kärkkäinen and Sanders, 2003). There is one caveat regarding the memory requirement here: the suffix array for a sequence of length l contains a permutation of the first l integers; while the length of this array is linear in l, the number of digits required to specify each integer grows logarithmically with l as well. An uncompressed suffix array (as used here) thus technically requires memory specified in bits scaling with $llogl$. Assuming the default use of 64-bit integers (as is done in the numpy-based python implementation we have used), however, memory will scale linearly for sequences of length up to $\sim 10^{18}$ characters, far beyond current practical limits.

Given the inverted suffix array i_s yielding the value of the suffix array index i corresponding to the suffix array value s_i, the block array (Equation (3)) can be constructed in linear time and space (again in terms of the length of the concatenated sequence x) by (1) looping through the positions s in the concatenated string x, (2) checking whether the active block b needs to be incremented according to whether $s \geq l_{b+1}$ (Section 2.1), and (3)

filling in the position i_s of the block array with the active block value b.

Kernel smoothing using a uniform kernel may be implemented in linear time by computing differences of cumulative sums (Equation (6)). The array of Gini impurity values (Equation (S4)) can be computed in linear time by successively computing the difference in consecutive values resulting from shifting the smoothing window by one position and updating the associated block frequencies Equation (S3). Identification of peaks (by comparing the score of each position to the scores of the two spatially adjacent positions) in the array of smoothed suffix scores $\hat{y}_i$, along with the filtering of the resulting peak set based on score threshold θ and Gini impurity threshold $g_{\min}$, again requires time linear in the length of the concatenated sequence x. Similar remarks hold for the analogous spatial smoothing operations.

Permutation testing requires repetition of the above steps R times, where R is the number of permutations, and is hence still asymptotically linear in the length of the concatenated sequence x. While in principle parallelizable, each permutation will require its own smoothed (and, if desired, spatially smoothed) score array, so that parallelization requires memory linear in $R * |x|$.

Motif length selection according to Equation (7) could be naively implemented in $O(k_{\max} * \kappa)$ time per peak by directly comparing each suffix in the smoothing window to the suffix corresponding to the suffix array index around which the window is centered. In fact it is generally faster to use the suffix array to compute the suffix array index bounds for which the k-mer prefixing the central suffix is conserved (this may be done quite efficiently using the Burrows-Wheeler transform (Ferragina and Manzini, 2000); in our implementation of SArKS we have generally avoided this in order to reduce the memory requirements of the algorithm, favoring instead a slightly less efficient binary search approach). Either way, motif length selection generally requires time linear in the size of the peak set; in practice, when the peak set is large, this step can be relatively time consuming.

Merging of spatially adjacent k-mers originating within the same spatial smoothing window (Equation (S15)) may be computed in time linear in the size of the peak set times the length of the spatial smoothing window λ. In the case of large peak sets, this step can be time consuming as well.

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 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).

References

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**.

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 setsupdate. *Nucleic Acids Research*, **41**(D1), D991–D995.

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*, page gkw334.

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

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

Close, J. L., Yao, Z., Levi, B. P., Miller, J. A., Bakken, T. E., Menon, V., Ting, J. T., Wall, A., Krostag, A.-R., Thomsen, E. R., *et al.* (2017). Single-cell profiling of an in vitro model of human interneuron development reveals temporal dynamics of cell type production and maturation. *Neuron*, **93**(5), 1035–1048.

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

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

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

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.

Ferragina, P. and Manzini, G. (2000). Opportunistic data structures with applications. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 390–398. IEEE.

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

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

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

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

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

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.

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*, page gkv1176.

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.

Reid, J. E. and Wernisch, L. (2014). STEME: a robust, accurate motif finder for large data sets. *PloS one*, **9**(3), e90735.

Saito, K. and Cui, H. (2018). Emerging roles of estrogen-related receptors in the brain: Potential interactions with estrogen signaling. *International Journal of Molecular Sciences*, **19**(4), 1091.

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.

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.

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