Supplementary Material

Contents

S1. Description of ChIP-seq data preprocessing

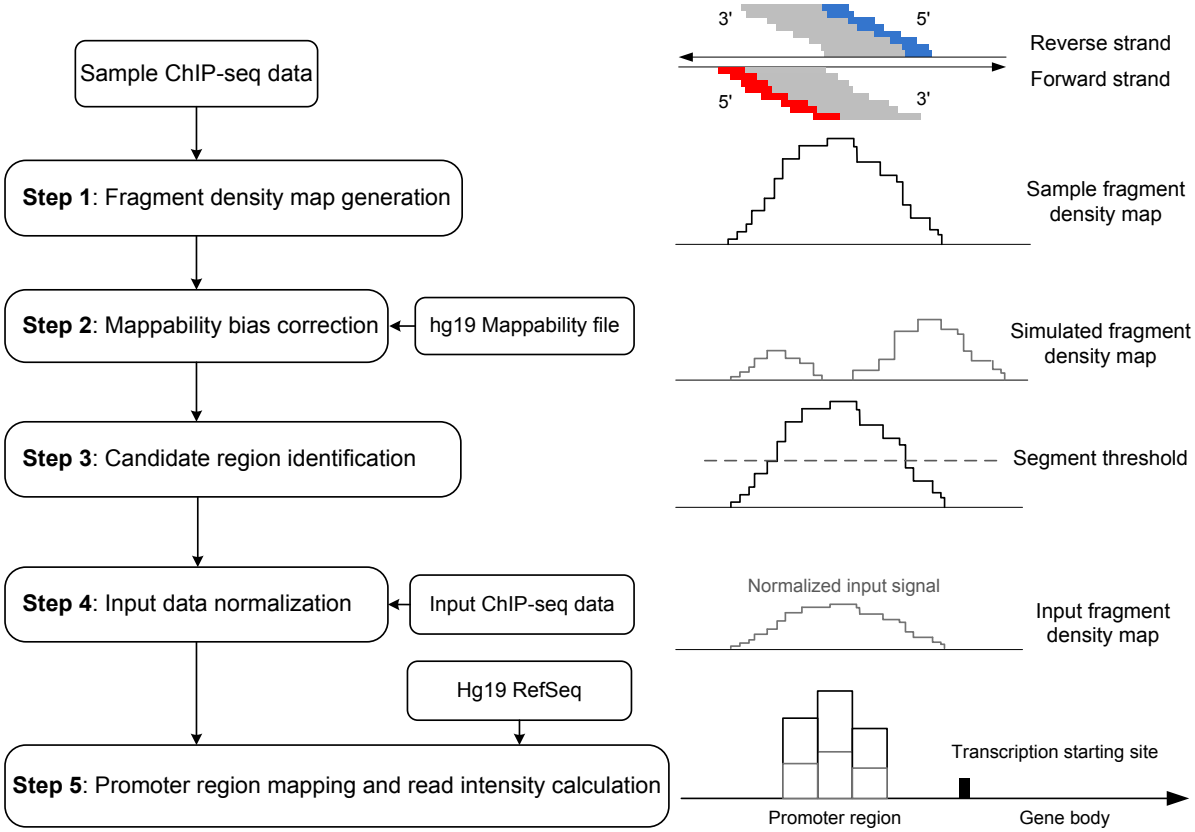


Fig. S1. Flowchart of data preprocessing.

A flowchart of the steps of data preprocessing is shown in Fig. S1, which are described in detail as follows:

Step 1: 5' locations of uniquely aligned reads (stored in BAM format) from a TF-DNA binding profile (sample ChIP-seq data) are extracted and further extended by the average DNA fragment length (200 bps) towards 3' direction. Fragments are then accumulated to form a fragment density map, as shown in Fig. S1. At nucleotide i, its fragment density can be calculated by counting forward reads with 5' start locations falling in $[i-200+1,i]$ and reverse reads with 5' start locations falling in $[i,i+200-1]$. At this step, ChIP-seq reads are extended and clustered together to form continues regions if they overlap with each other. Note that only uniquely mapped reads of sample ChIP-seq data are used in this step so the mappability bias at different locations of genome needs to be properly addressed before candidate region identification.

Step 2: Once the sample fragment density map has been created, each chromosome is partitioned into segments for candidate region searching according to a mappability map with segmentation length l. 'Mappbability' is defined as the number of uniquely mappable nucleotides in a genome segment with length l. The default value of l is set as 1M bps in PeakSeq (1). For each segment, its uniquely mappable fraction (mappability/l) is denoted as f (0~1), and its fragment count is denoted as N. In Fig. S2(a), we present f values for five 1M segments of human genome (hg19) chr1:1~5,000,000 (PBX1 ChIP-seq data). A histogram of N of all segments from the whole genome is shown in Fig. S2(b). It can be seen from Fig. S2(a) that the variation of f is not significant or very sensitive for different segments especially among those segments with high f values.

As mentioned by Kuan *et al.* (2), l should be as small as 1k to make mappability variation more apparent, as shown in Fig. S2(e). However, Kuan *et al.* also pointed out that at 1k scale there may not be enough reads to build the background model (by performing random permutations of reads). As can be seen from Fig. S2(f), most segments only have less than 10 reads, which are too few to build the background model. In ChIP-BIT, we focus on gene promoter regions defined as relative ±10k bps from the transcription starting site. Therefore, for our specific problem, we set the segment length l as 20k. As shown in Fig. S2(c), the sensitivity of this segment length is comparable to Fig. S2(e) and much better than Fig. S2(a). It can be seen from Fig. S2(d) that a majority of segments have more than 50 reads, the number of which is suitable for building the null background model with random perturbations.

To correct any bias in N caused by f, we perform a computational simulation by randomly generating N fragments in a scaled segment of length $f \times l$. Using a height threshold we can determine all the contiguous regions that are above this threshold in the sample fragment density map. Regions above threshold are merged together if their genomic distances are less than the average fragment length (200 bps). For the same threshold we can determine the number of regions above the threshold in the simulated fragment density map as well. In each segment, the false discovery rate (FDR) of a selected height threshold can be calculated as follows:

$$FDR(threshold) = \frac{(\#\text{ regions above threshold in simulation})}{(\#\text{ regions above threshold in the ChIP-seq sample})}$$

To meet a target FDR requirement, e.g., 0.05, a height threshold is selected independently for each segment of each chromosome, which accounts for genomic variability along each segment.

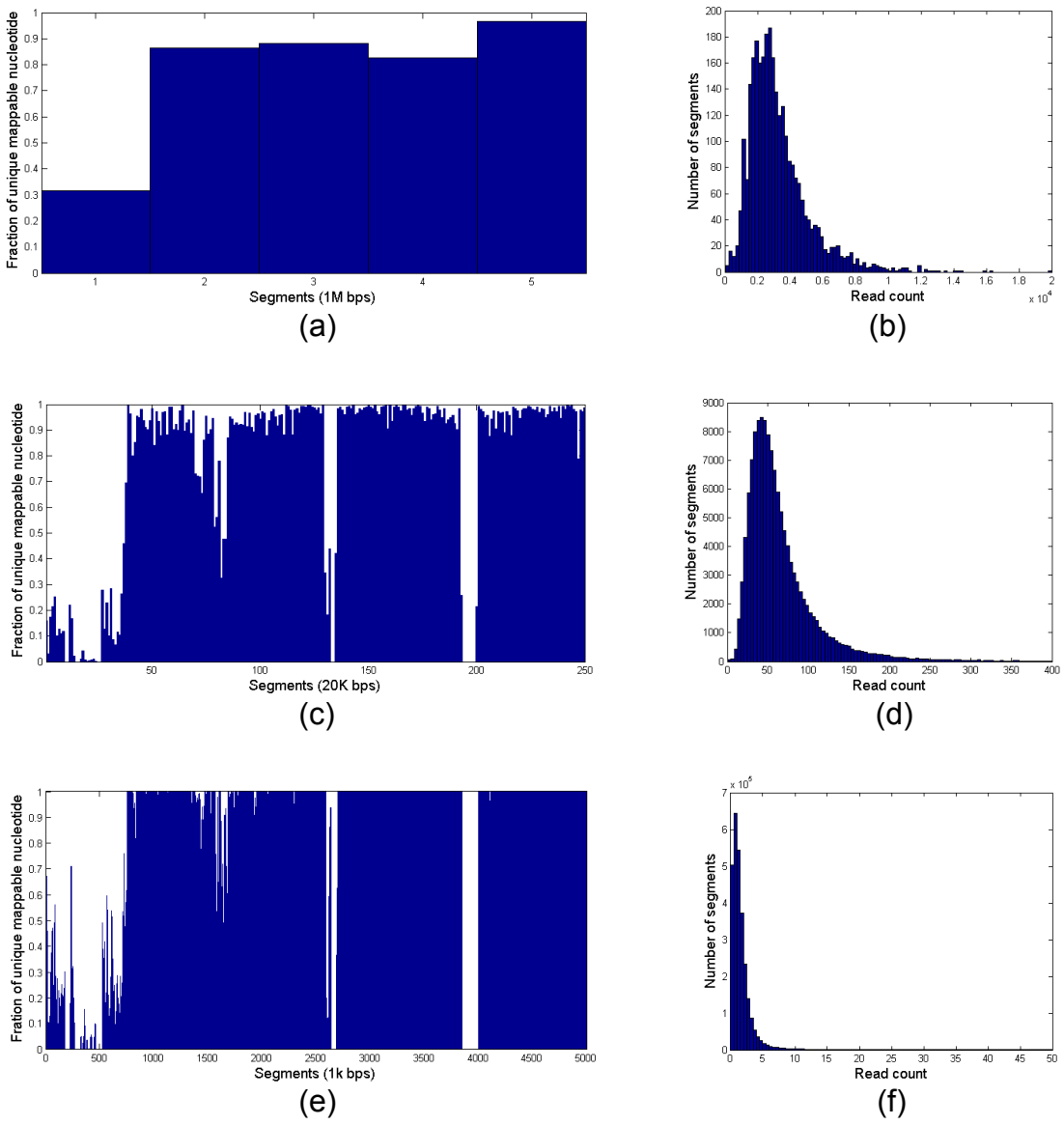


Fig. S2. Genome mappability and read count distributions using different segmentation lengths: (a) genome mappability of 1M segmentation; (b) read count distribution (1M bps); (c) genome mappability of 20k segmentation; (d) read count distribution (20k bps); (e) genome mappability of1k segmentation; (f) read count distribution (1k bps).

Step 3: From each segment, a list of candidate regions with height larger than the minimum threshold meeting the FDR requirement is obtained. Candidate regions from all segments are collected together. For each region, uniquely mapped reads in the control profile (Input ChIP-seq data) are then counted. Note that candidate regions are highly enriched in the sample profile but may not be significant if compared to the local input signal. Therefore, after this step, the candidate pool actually includes both peak and background regions. Further peak calling is needed for true peak detection.

(Note: Steps 1 - 3 in our data preprocessing are performed using the first pass (ChIP-seq data preprocessing) of the PeakSeq package (version 1.31).)

Step 4: Read counts of the control profile are further normalized against those counts in background regions of the sample data. Since it is generally not applicable to directly determine the proportion of background regions out of all candidate regions in the sample data, all enriched regions in the sample data are sorted according to their read counts from low to high. We define ρ as a quantile threshold to select low read count regions for data normalization between sample and input data. By changing ρ from 0 to 1 with a step size of 0.2, we calculate a set of regression coefficients using linear regression of read counts in selected segments between the sample and the input data. The scaling factor uses the median value of all regression coefficients calculated. We amplify input read count using the scaling factor and exclude any regions in our candidate pool whose input read count is larger than that of sample data.

Step 5: We map candidate regions to partitioned windows (200 bps) at gene promoter regions. Read intensities are calculated as the natural log value of accumulated read coverage in each window of the sample and input data, respectively. More details about read intensity calculation will be given later in Section S3.1. While there are no '0' coverage regions in the sample data (after the processing of **Step 1**), some regions may have '0' coverage in the input data. For these regions, we use their coverage as the minimal value of non-zero converges of all regions and then apply the log transformation. For the promoter region mapping, we take a similar strategy as GREAT (3) or TIP (4). First, every gene is assigned a promoter region as ±10k bps from the TSS (regardless of other nearby genes). Each promoter region is then partitioned into 200 bps long windows. Finally, each candidate region is associated with all genes whose promoter regions overlap it.

S2. Implementation of ChIP-BIT

A flowchart of the implementation of ChIP-BIT is illustrated in Fig. S3, and the formats of input and output files are listed in Table S1. In the following subsections, we will describe the implementation of the major steps (i.e., data format transformation, ChIP-seq data preprocessing and ChIP-BIT) in detail.

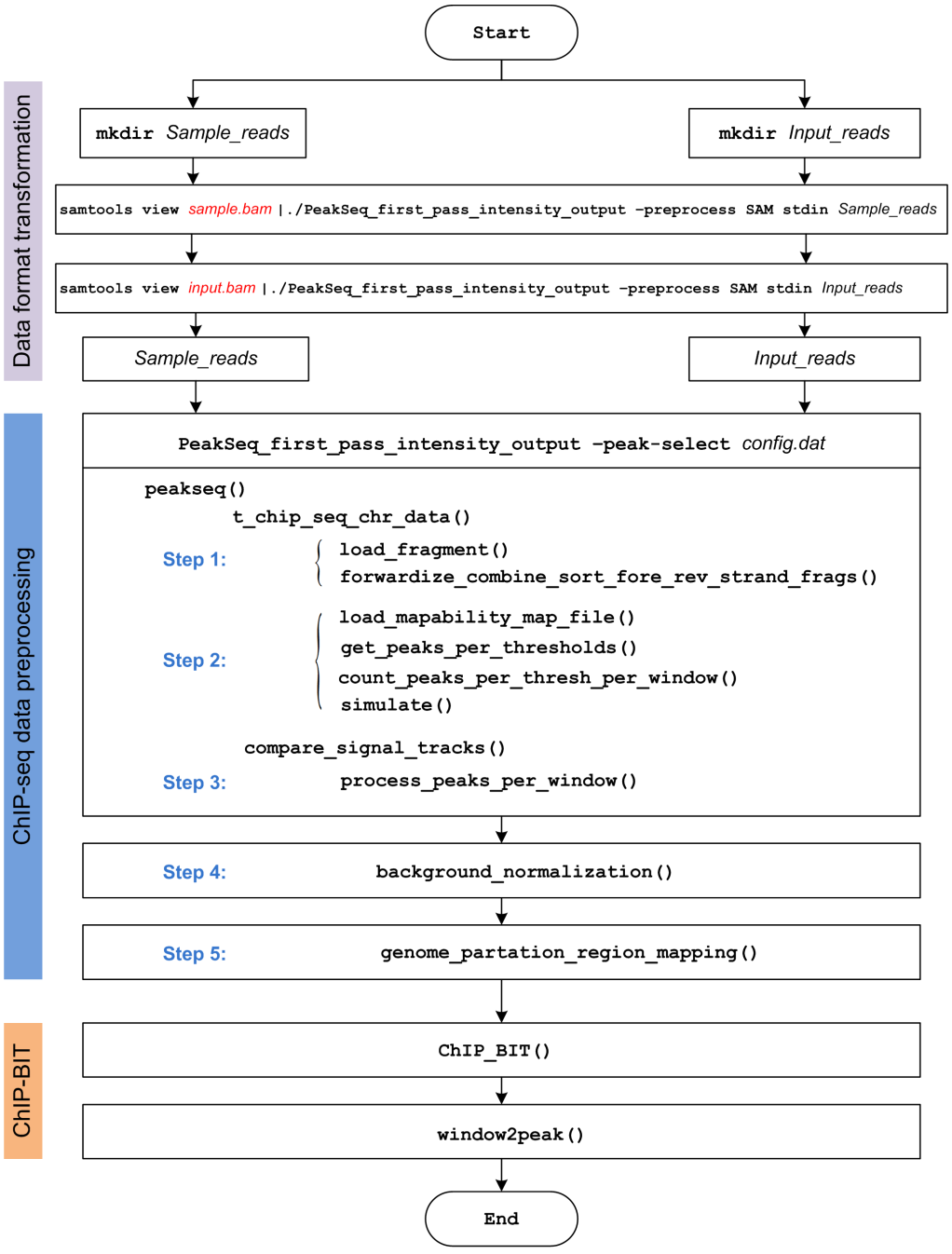


Fig. S3. Flowchart of the ChIP-BIT implementation.

Table S1. Input and output file formats of ChIP-BIT workflow

	File name	Format
Input files	Sample ChIP-seq data.bam	BAM
	Input ChIP-seq data.bam	BAM
	Hg19_mappability_20K.txt	TAB
	Hg19_RefSeg.txt	TAB
	Conifg.dat	Defined by PeakSeq
Output files	Candidate regions.txt	TAB
	ChIP-BIT_peaks.txt	TAB

S2.1 ChIP-seq data transformation

A folder “Sample_reads” (or “Input_reads”) is created for Sample (or Input) ChIP-seq BAM file to record 5’ start locations of reads for each chromosome. Sample or Input ChIP-seq data are dumped into the system using “samtools” and further processed using “PeakSeq_first_pass_intensity_output” with “-preprocess” option to report 5’ start locations of reads for each chromosome. If whole genome is tested, chromosomes 1 - 22, X, Y and M will be recorded separately with file name as “chrxx_mapped_reads.txt”, where read length, strand and 5’ location of each read tag are recorded as follows:

```
read length     strand     5’ location
…
36M             R          10095
36M             F          10100
36M             R          10101
36M             F          10102
36M             R          10103
36M             F          10106
…
```

S2.2 ChIP-seq data pre-processing

For ChIP-seq data preprocessing, we use a modified version of “PeakSeq”, “PeakSeq_first_pass_intensity_output”, to perform Steps 1-3 and provide two functions written in MATLAB or R language to perform Steps 4 and 5.

Step 1: We use function “t_chip_seq_chr_data()” to perform read extension and clustering. In detail, we use function “load_fragments()” to load forward and reverse read tags 5’ locations from folder “Sample_reads” and then, extend each read tag to a 200 bps long fragment and cluster forward or reverse fragments together if they form continuous regions using function “forwardize_combine_sort_fore_rev_strand_frags()”. Reads in “Input_reads” are also loaded to the system using function “load_fragments()”.

Step 2: In the function “t_chip_seq_chr_data()”, we use function “load_mapability_map_file()” to load mappability map for every 20k bps segment from file “hg19_mappability_20k.txt”, the format of which is shown as follows:

```
chr         index       number of unique nucleotide
1           0           3206
1           1           611
1           2           3453
1           3           4275
1           4           5034
1           5           2034
1           6           2558
…
```

Then, we vary the threshold from min_thresh to max_thresh and identify continuous regions using function “get_peaks_per_thresholds()”. For each segment, we count the number of continuous regions under each threshold using function “count_peaks_per_thresh_per_window()”. Finally, we generate a null background model and calculate a threshold meeting the target FDR requirement for each segment using function “simulate()”.

Step 3: In each segment, based on the calculated threshold from Step 2, we identify candidate regions using function “compare_signal_tracks()”, where function “process_peaks_per_window()” is used to report location and read enrichment information of each candidate region. A file ‘Candidate_regions.txt’ is created to store region information; see below for an example:

```
chr   start end   Sample_boudary   Sample_central   Sample_count
      Input_boudary    Input_central    Input_count
1  150601417  150602389  0.538  87.72   1793  0.026  4.256  87
1  151254049  151254918  0.762  92.764  1705  0.049  5.93   109
1  1207735    1208361    1.109  86.464  1169  0.045  3.476  47
1  110881197  110882220  0.513  56.897  1221  0.026  2.842  61
1  23405332   23406309   3.116  47.955  985   0.038  0.584  12
…
```

All above-mentioned functions in Steps 1-3 have been compiled into a binary file "PeakSeq_first_pass_intensity_output". A configuration file "Config.dat" is set as follows:

```
Experiment_id                           Candidate_regions
Enrichment_mapped_fragment_length       200
target_FDR                              0.05
N_Simulations                           10
Minimum_interpeak_distance              200
Mappability_map_file                    hg19_mappability_20k.txt
ChIP_Seq_reads_data_dirs                Sample_reads
Input_reads_data_dirs                   Input_reads
Background_model                        Simulated
```

Step 4: Before peak calling, a scaling factor is calculated using function "background_normalization()" to normalize input ChIP-seq data to the similar scale of sample ChIP-seq data. Both MATLAB and R versions of this function are provided in the software package.

Step 5: We load gene promoter regions from file 'hg19_RefSeq.txt'. The gene annotation file can be downloaded from the UCSC Genome Browser (https://genome.ucsc.edu/). The format of the annotation file can be seen as the following:

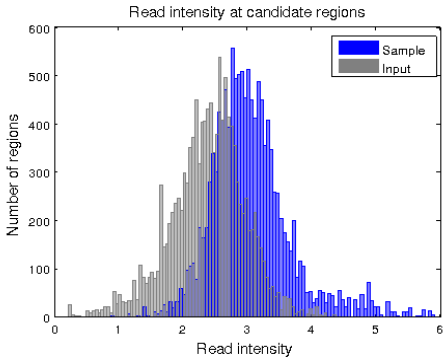
```
chr         strand      txStart     txEnd       gene_symbol
chr1        -           33772366    33786699    A3GALT2
chr1        +           12776117    12788726    AADACL3
chr1        +           12704565    12727097    AADACL4
chr1        -           94458393    94586705    ABCA4
…
```

Then, we use function "genome_partation_region_mapping()" to partition each gene promoter regions into non-overlapping windows, map candidate regions to windows and for each window calculate read intensities respectively in sample and input ChIP-seq data. Both MATLAB and R versions of this function are provided in the software package. An example of the output is shown below:

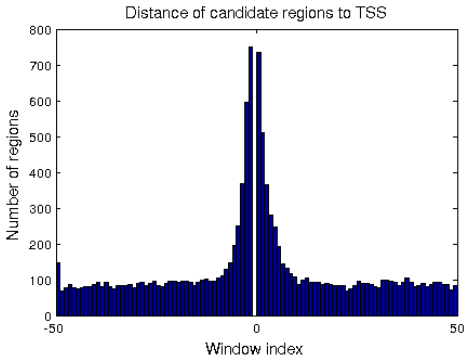
```
chr    start      end   sample_intensity      input_intensity  direction
window_index      gene_symbol
'1'   33776700   33776899   2.345      1.862 '-'   50    'A3GALT2'
'1'   33776900   33777099   2.345      1.862 '-'   49    'A3GALT2'
'1'   33779900   33780099   2.439      1.407 '-'   34    'A3GALT2'
'1'   33780900   33781099   2.469      1.688 '-'   29    'A3GALT2'
'1'   33781100   33781299   2.469      1.688 '-'   28    'A3GALT2'
'1'   33786900   33787099   2.615      2.260 '-'   -2    'A3GALT2'
'1'   12772918   12773117   2.099      0.779 '+'   -16   'AADACL3'
'1'   12773118   12773317   2.099      0.779 '+'   -15   'AADACL3'
```

S2.3 ChIP-BIT peak detection

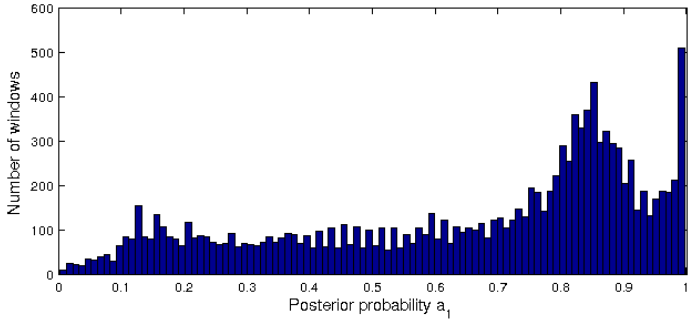
The core function for peak detection is accomplished by function “ChIP_BIT()”. For each candidate window, we estimate a posterior probability for binding occurrence. Histograms of read intensity, window index (relative distance to TSS), and posterior probability for all candidate windows are shown in Fig. S4(a), (b) and (c), respectively. Both MATLAB and R versions of this function are provided in the software package.



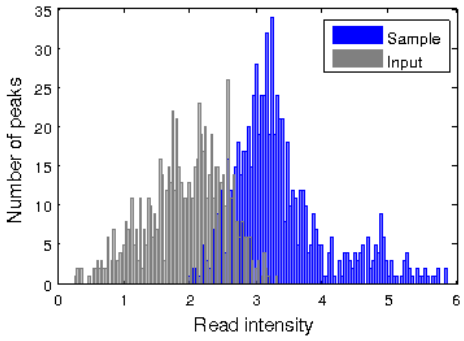
(a) Read intensity of candidate windows



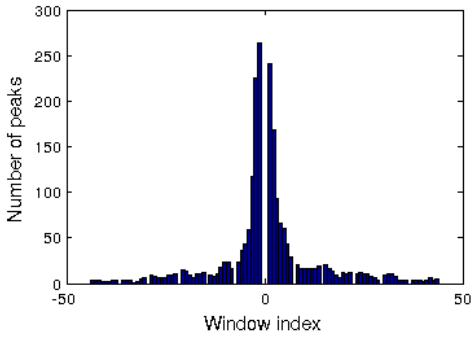
(b) Window index of candidate windows



(c) Posterior probabilities for binding occurrence



(d) Read intensity of detected peaks



(e) Relative distance of detected peaks

Fig. S4. Histograms of input and output signals in ChIP-BIT peak detection.

Windows with posterior probabilities over 0.90 are selected and consecutive windows are further merged into larger peaks by function “window2peak()”. Both MATLAB and R versions of this function are provided in the software package. Histograms of read intensity and window index for detected peaks are shown in Fig. S4(d) and (e), respectively.

Finally, a peak file 'ChIP_BIT_peaks.txt' is created to store all peaks detected by ChIP-BIT. Please see below for an example:

chr	start	end	gene_symbol	Sample_Reads_intensity	Input_Reads_intensity	Posterior_probability
1	229694243	229695242	ABCB10	3.140179	2.156729	0.947231
1	1243270	1243869	ACAP3	3.216313	2.193865	0.950682
1	226374224	226375023	ACBD3	3.266675	2.092388	0.967518
1	159169803	159170002	ACKR1	3.133405	1.659656	0.974730
1	55012807	55013806	ACOT11	2.934070	2.106528	0.967670
1	6457027	6457426	ACOT7	3.082827	1.762697	0.935282
1	120434948	120435347	ADAM30	3.061520	1.134501	0.983762
1	167883865	167884064	ADCY10	3.293018	1.819366	0.992985
1	202928301	202928500	ADIPOR1	3.521880	2.694488	0.905599
1	244616637	244623836	ADSS	3.488231	1.762697	0.954942
1	50488427	50488626	AGBL4	2.953868	1.479771	0.991095
1	15911006	15911405	AGMAT	2.763043	1.577070	0.975546
…						

Note that source codes of the implementation of ChIP-BIT and a pair of testing PBX1 and input ChIP-seq data sets (with chromosome 1 only) are provided at http://www.cbil.ece.vt.edu/software.htm. More details about the use of each function can be found in the user manual of ChIP-BIT in the package.

S3. Distribution hypothesis on read intensity and relative distance

To demonstrate the distribution hypotheses on read intensity and relative distance to TSS of ChIP-seq binding events, we use PBX1 under MCF7 cell line (GSE accession number GSE28008) as an example to examine real ChIP-seq data. Extra ChIP-seq data from TFs under different conditions are analyzed as well. Note that regions used here for each TF are all candidate regions including peaks and background regions, as identified before peak calling (processed after **Step 5** in Fig. S1).

S3.1 Read intensity calculation

N_{region} is defined as the total number of forward and reverse reads falling in each candidate region, as illustrated in Fig. S5(a). We present the histogram of read count N_{region} for all candidate regions in Fig. S6(a). It can be expected that a wider region is more possible to contain a relatively larger N_{region}. Therefore, it is not fair to directly evaluate read enrichment across different regions when their lengths are very different. The fragment density map shown in the lower part of Fig. S5(a) is very similar to the ChIP-seq coverage information stored in WIG format. However, no read count information can be recovered if WIG format ChIP-seq profile is directly used. In that case, the genome mappability variation across different segments cannot be assessed. This is the major reason why we start from BAM format ChIP-seq data.

Read intensity s for a 200 bps long window is defined as the natural log transformation of the accumulated read coverage C_{200}. The reason we select window size as 200 is due to the observations of significant TF peak length distribution (peaks identified by PeakSeq with default setting), as shown in Fig. S5(b).

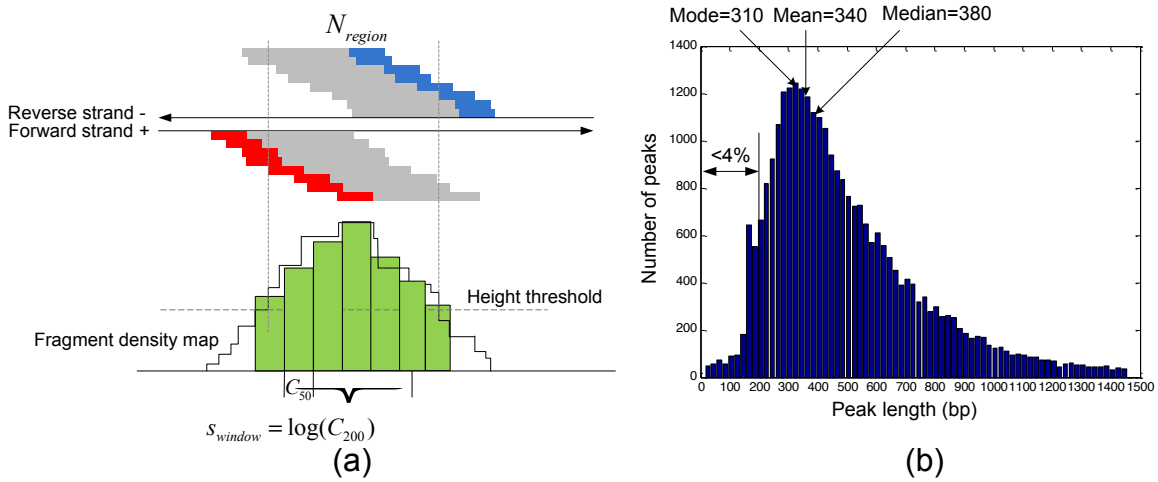


Fig. S5. Illustrations of read count, coverage and peak length for TF peaks: (a) 50 bps bin decomposition of each region; (b) peak length distribution.

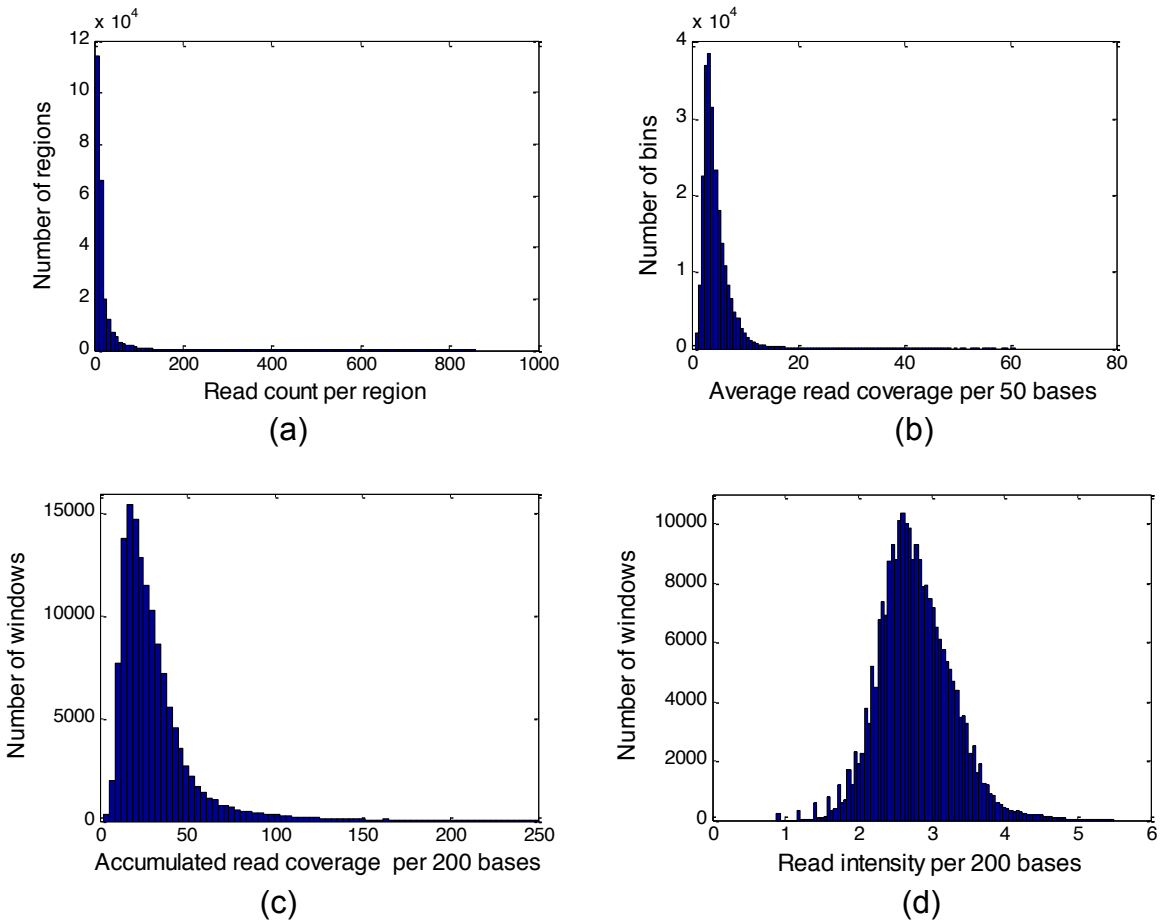


Fig. S6. Read intensity calculation: (a) read count per region; (b) read coverage per 50 bps; (c) overall read coverage per 200 bps; (d) read intensity per 200 bps.

As can be seen from Fig. 5(b), there are quite few peaks with length less than 200 bps (<4%). And the mean (340 bps), median (380 bps) and mode (310 bps) of peak length all fall between 200 and 400. Therefore, each peak will occupy at least one window and most peaks will cover 2 or 3 windows. If the fragment density C_1 at each base is known, the accumulated read coverage C_{200} for each window can be calculated as $C_{200} = \sum_{i=1}^{200} C_{1,i}$, where i is the location index of each base of each window. However, fragment density information like C_1 is very redundant considering that a typical read length for current ChIP-seq platform (i.e. illumina HiSeq 2500) is 36 or 50 bps and each read will be extended to a ~150-200 bps long fragment for further analysis.

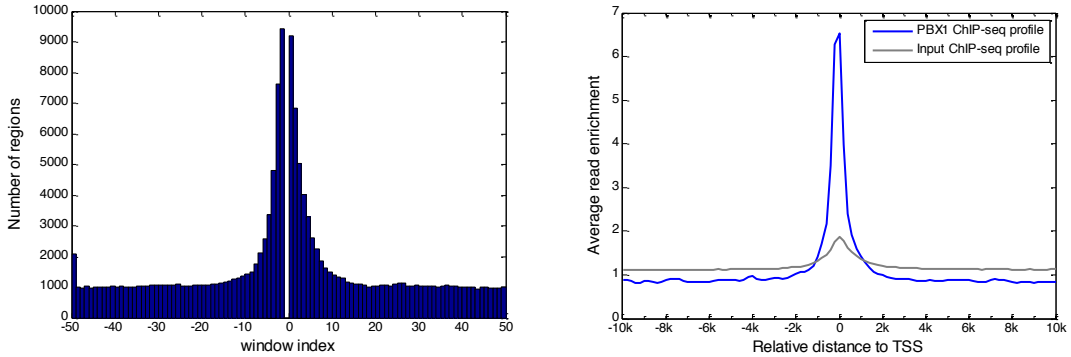
A reasonable length of a bin used to model peaks of a TF ChIP-seq profile is 50 bps (as used by MOSAiCS). Based on the fragment density map in Fig. 5(a), we can calculate the average read coverage C_{50} for every bin (green bar in Fig. S5(a)). C_{50} takes the average value of fragment density at each covered location as $\sum_{i=1}^{50} C_{1,i} / 50$ and its histogram is shown in Fig. S6(b). In practical implementation, for each region we

calculate C_{50} values at central 50 bps (±25 bps around the summit) and boundary 50bps (±25 bps around the boundary point with a height that equals to the threshold of current segment) respectively. For any bins between central and boundary locations, we estimate a value C_{50} by assuming that the read coverage linearly decreases from central location to boundary location. Then, $\hat{C}_{200}$ can be approximated by $\sum_{i=1}^{4} C_{50,i}$. The major difference between C_{200} and $\hat{C}_{200}$ is a constant amplification factor 50, which is the same for all regions. The histogram of $\hat{C}_{200}$ is shown in Fig. S6(c). Finally, we calculate the read intensity by $s = \log(\hat{C}_{200})$; the histogram of *s* can be found in Fig. S6(d), where a Gaussian distribution can be clearly observed. Read intensity of input data is calculated in the same way for each candidate region.

S3.2 TF binding location analysis

We first check the number of windows overlapping with candidate regions of PBX1 at gene promoter regions and present the histogram in Fig. S7(a) based on their locations (window index), where an exponential distribution feature can be seen. Mokry *et al*. (5) used a similar way to evaluate binding preference of TF at gene promoter regions (±200k bps).

Users/researchers may have some concern that this exponential distribution is caused by mappbability variation or some other bias issues. Such bias issues are features of genome, independent of sample or input experiment. Therefore, for one TF, if a different feature can be found between sample and input ChIP-seq data, it is highly possible to be caused by TF binding events. We use a function “annotatePeaks.pl” of an independent package HOMER (6) to directly evaluate the average ChIP-seq read count in each window (200 bps) across all promoter regions for sample and input ChIP-seq profiles, respectively. Similar work was also carried out in (7) to evaluate binding features of 12 TFs ChIP-seq data sets where regions of ±4k from TSS were examined.



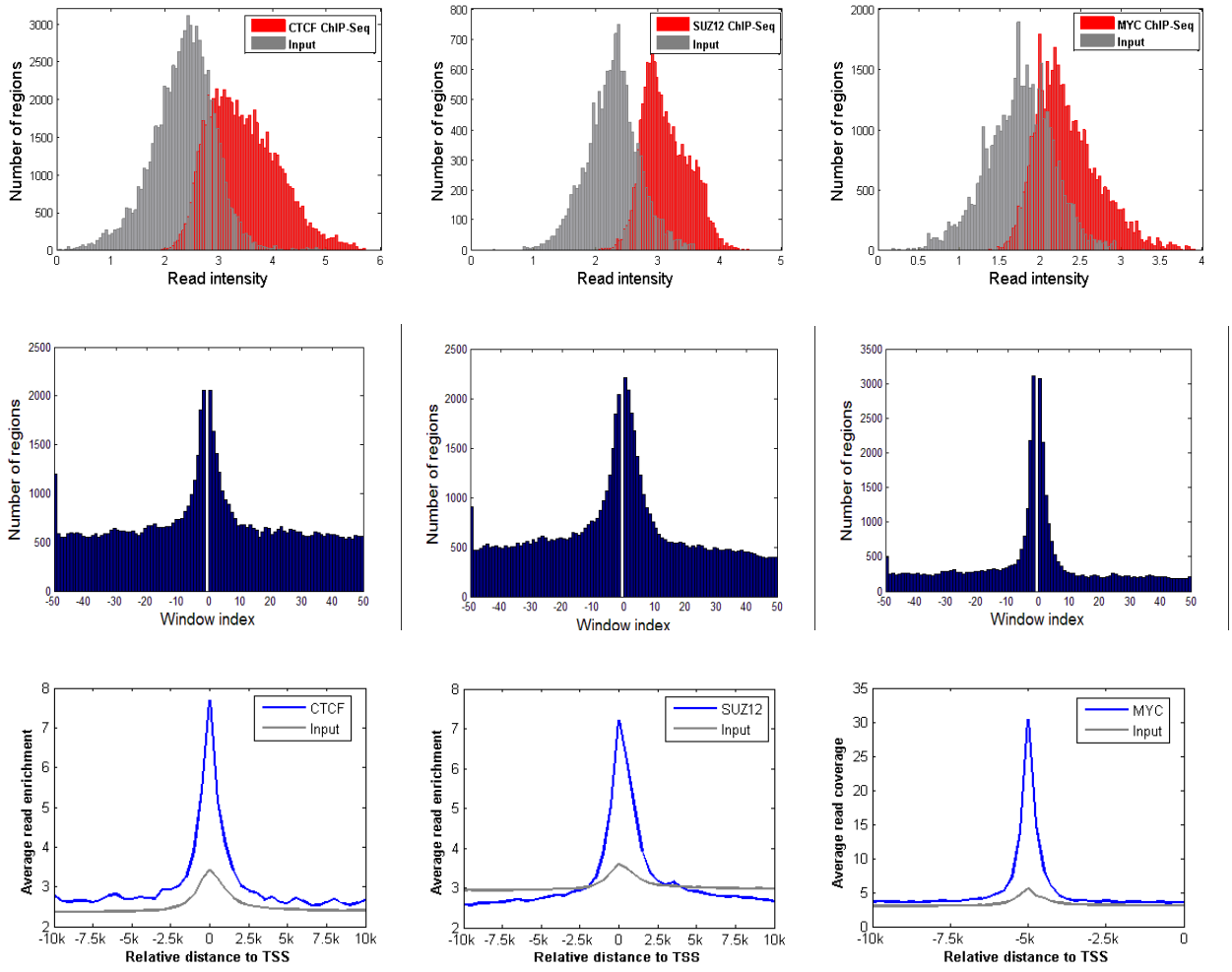
(a) (b)

Fig. S7. PBX1 binding at gene promoter regions: (a) number of regions in each window; (b) average read enrichment of each window.

As shown in Fig. S7(b), there is an exponential distribution in the sample ChIP-seq data while a more uniform distribution in the input. The slight increase of input read count distribution around TSS would be possible to be caused by repetitive sequences. But compared to the sample data amplitude at the same location, it is much lower. All above observations validate our hypothesis on location-wise distribution of TF ChIP-seq data enrichment at gene promoter regions.

S3.3 ChIP-seq read enrichment and binding location under different conditions

Observed features about read intensity and binding location distributions are further checked for multiple TFs under different conditions, as shown in Fig. S8. The sample and input ChIP-seq data is downloaded from https://genome.ucsc.edu/ENCODE/.



(a) (b) (c)

Fig S8. Distributions of read intensity and relative distance to TSS of candidate regions for each selected TF: (a) CTCF of MCF7 cell line; (b) SUZ12 of H1-hESC cell line; (c) MYC of K562 cell line.

S4. Probability mass function of the distance from the TFBS to the TSS

The conditional probability $P(d_{n,w} \mid b_{n,w})$ for a binding site at w-th window of the promoter region of n-th gene is determined by both its relative distance $d_{n,w}$ and its binding state $b_{n,w}$. If $b_{n,w} = 1$, the binding effect decays exponentially when $d_{n,w}$ increases. Thus, the conditional probability $P(d_{n,w} \mid b_{n,w} = 1)$ is defined by a discrete exponential distribution as follows:

$$P(d_{n,w} \mid b_{n,w} = 1) = \frac{1}{C_d} \exp\left(-\lambda \left| d_{n,w} \right|\right), \tag{S-1}$$

where $d_{n,w} = (w + sign(w) * \frac{1}{2})\Delta d$, $w = \{0, \pm 1, \pm 2, ..., \pm(W-1)\}$ is the index of window location and Δd is the window size. Relative distance $d_{n,w}$ is the relative distance from the midpoint of each window to the TSS. Please see Fig. S9 for an illustration of Eq. (S-1).

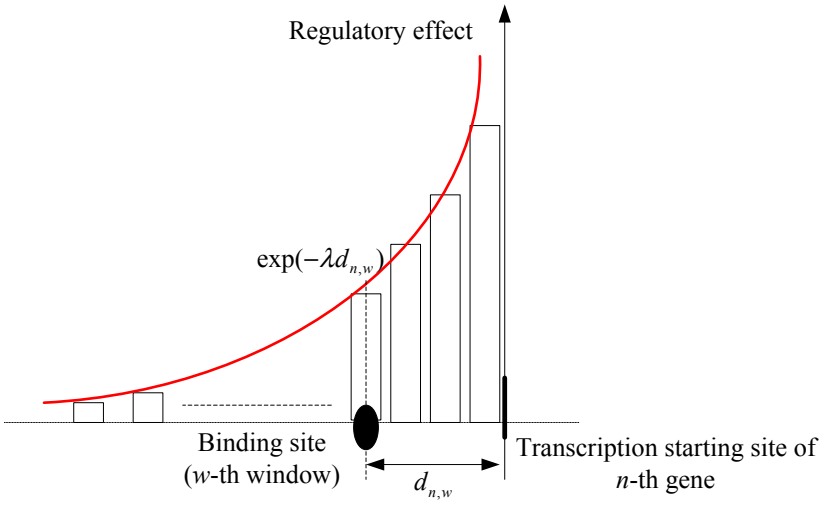


Fig. S9. Regulatory effect of binding site locations to the TSS of target genes.

Since we are using a discrete exponential distribution in (S-1), C_d is a normalized factor that can be calculated as:

$$C_d = \sum_{w=-(W-1)}^{W-1} \exp\left(-\lambda(|w| + 1/2)\Delta d\right) = 2\exp(-\frac{\lambda \Delta d}{2}) \frac{1 - \exp(-\lambda W \Delta d)}{1 - \exp(-\lambda \Delta d)}. \tag{S-2}$$

Considering that $W\Delta d = 10k$ and $\lambda W\Delta d >> 1$, we have $\exp(-\lambda W\Delta d) \to 0$. We bring $C_d = 2\exp(-\frac{\lambda\Delta d}{2}) / (1-\exp(-\lambda\Delta d))$ back to (S-1) and the conditional probability $P(d_{n,w} \mid b_{n,w} = 1)$ can be finally calculated as:

$$P(d_{n,w} \mid b_{n,w} = 1) = \frac{1}{2}\exp\left(-\lambda|w|\Delta d\right)\left(1-\exp(-\lambda\Delta d)\right). \quad \text{(S-3)}$$

For any background region with binding state $b_{n,w} = 0$, its distance to the TSS is non-informative for its regulatory effect(s) on target gene(s). Therefore, we define the conditional probability $P(d_{n,w} \mid b_{n,w} = 0)$ as a uniform distribution on $d_{n,w}$.

S5. EM-based posterior probability estimation

The posterior probability used to estimate variables is defined as:

$$\begin{aligned} P(\pi, \mu_{TFBS}, \sigma^2_{TFBS}, \lambda \mid \mathbf{s}, \mathbf{d}) &= \frac{1}{C_1}\prod_n\prod_w P(s_{n,w}, d_{n,w} \mid \mu_{TFBS}, \sigma^2_{TFBS}, \lambda, \pi)\frac{1}{C_\mu}\frac{1}{C_\lambda}P(\pi)\cdot P(\sigma^2_{TFBS}) \\ &= \frac{1}{C_2}\prod_n\prod_w\sum_{b_{n,w}=0,1}\left[P(s_{n,w} \mid b_{n,w})P(d_{n,w} \mid b_{n,w})P(b_{n,w})\right]\cdot P(\pi)\cdot P(\sigma^2_{TFBS}) \end{aligned} \quad \text{. (S-4)}$$

By introducing Jensen's inequality with $\sum_{i=0,1}\hat{a}_{n,w,i} = 1$, we obtain the following inequality:

$$\begin{aligned} -\log P(\pi, \mu_{TFBS}, \sigma^2_{TFBS}, \lambda \mid \mathbf{s}, \mathbf{d}) &= -\sum_n\sum_w \log\left(\sum_{b_{n,w}=0,1} P(s_{n,w} \mid b_{n,w})P(d_{n,w} \mid b_{n,w})\pi^{b_{n,w}}(1-\pi)^{(1-b_{n,w})}\right) \\ &\quad -\log P(\boldsymbol{\pi}) - P(\sigma^2_{TFBS}) + \log C_2 \\ &\le -\sum_n\sum_w\sum_{b_{n,w}=0,1}\hat{a}_{n,w}\log\frac{1}{\hat{a}_{n,w}}P(s_{n,w} \mid b_{n,w})P(d_{n,w} \mid b_{n,w})\pi^{b_{n,w}}(1-\pi)^{(1-b_{n,w})} \\ &\quad -\log P(\boldsymbol{\pi}) - P(\sigma^2_{TFBS}) + \log C_2 \end{aligned} \quad \text{. (S-5)}$$

Maximizing (S-4) is equivalent to minimizing the upper bound of its natural log format as shown in (S-5) and supports our ability to estimate hidden variables (8). The detailed steps of EM algorithm can be summarized as follows (where Step 1 is the E-step and Steps 2-5 are the substeps of the M-step):

Step 1: Estimate $\hat{a}^*_{n,w,i}$

To minimize the upper bound of the right part of (S-5), we seek $\hat{a}_{n,w,i}$ to meet the equality condition of (S-5) as follows:

$$\log\left[\frac{1}{\hat{a}_{n,w,i}}P(s_{n,w}\mid b_{n,w}=i)P(d_{n,w}\mid b_{n,w}=i)\pi_i\right]=\log\left[\sum_i P(s_{n,w}\mid b_{n,w}=i)P(d_{n,w}\mid b_{n,w}=i)\pi_i\right]. \quad \text{(S-6)}$$

Therefore, $\hat{a}^*_{n,w,i}$ can be estimated as

$$\hat{a}^*_{n,w,i}=\frac{P(s_{n,w}\mid b_{n,w}=i)P(d_{n,w}\mid b_{n,w}=i)\pi_i}{\sum_i P(s_{n,w}\mid b_{n,w}=i)P(d_{n,w}\mid b_{n,w}=i)\pi_i}. \quad \text{(S-7)}$$

Step 2: Estimate π^*_i

Extending (S-5) and excluding items independent of π_i, we obtain $f(\boldsymbol{\pi})$ as

$$f(\boldsymbol{\pi})=-\sum_n\sum_w\sum_i \hat{a}_{n,w,i}\log\pi_i-\log P(\boldsymbol{\pi}),\ \sum_i\pi_i=1. \quad \text{(S-8)}$$

Minimizing the upper bound of (S-5) is equivalent to finding π_i where $\frac{\partial f(\boldsymbol{\pi})}{\partial \pi_i}=0$. To address the constraint of $\sum_i\pi_i=1$, we introduce a Lagrange parameter τ to $f(\boldsymbol{\pi})$.

$$\begin{aligned} f(\boldsymbol{\pi}) &= -\sum_n\sum_w\sum_i \hat{a}_{n,w,i}\log\pi_i-\log P(\boldsymbol{\pi})+\tau(\sum_i\pi_i-1) \\ &= -\sum_n\sum_w\sum_i \hat{a}_{n,w,i}\log\pi_i-\sum_i(\beta_i-1)\log\pi_i+\log B(\beta)+\tau(\sum_i\pi_i-1) \end{aligned}. \quad \text{(S-9)}$$

The derivative of (S-9) in term of π_i is computed as

$$\frac{\partial f}{\partial \pi_i}=-\sum_n\sum_w \hat{a}_{n,w,i}\frac{1}{\pi_i}-(\beta_i-1)\frac{1}{\pi_i}+\tau=0. \quad \text{(S-10)}$$

Since $\sum_i\pi_i=1$ and $\sum_i\hat{a}_{n,w,i}=1$, we sum (S-10) using the value(s) of i and obtain $\tau=T+\beta_1+\beta_2-2$.

Bring τ back to (S-10), we can estimate π^*_i as

$$\pi^*_i = \frac{\sum_n \sum_w \hat{a}^*_{n,w,i} + (\beta_i - 1)}{T + \beta_1 + \beta_2 - 2}. \quad \text{(S-11)}$$

Step 3: Estimate μ^*_{TFBS}

Extending (S-5) and excluding items independent on μ_{TFBS} , we obtain $f(\mu_{TFBS})$ as

$$f(\mu_{TFBS}) = -\sum_n \sum_w \left[\hat{a}_{n,w,1} \left(\log \frac{1}{\sqrt{2\pi}\sigma_{TFBS}} - \frac{1}{2} \left(\frac{s_{n,w} - \mu_{TFBS}}{\sigma_{TFBS}} \right)^2 \right) \right]. \quad \text{(S-12)}$$

Minimizing the upper bound of (S-5) is equivalent to finding μ_{TFBS} where $\frac{df(\mu_{TFBS})}{d\mu_{TFBS}} = 0$.

The derivative of (S-12) in terms of μ_{TFBS} is computed as

$$\frac{df(\mu_{TFBS})}{d\mu_{TFBS}} = \sum_n \sum_w \hat{a}_{n,w,1} \frac{1}{\sigma^2_{TFBS}} \left(\mu_{TFBS} - s_{n,w} \right) = 0. \quad \text{(S-13)}$$

Finally, we can estimate μ^*_{TFBS} as

$$\mu^*_{TFBS} = \frac{\sum_n \sum_w \hat{a}_{n,w,1} s_{n,w}}{\sum_n \sum_w \hat{a}_{n,w,1}}. \quad \text{(S-14)}$$

Step 4: Estimate σ^*_{TFBS}

Extending (S-5) and excluding items independent on σ_{TFBS} , we obtain $f(\sigma_{TFBS})$ as

$$\begin{aligned} f(\sigma_{TFBS}) &= -\sum_n \sum_w \hat{a}_{n,w,1} \log P(s_{n,w} \mid b_{n,w} = 1, \mu_{TFBS}, \sigma^2_{TFBS}) - \log P(\sigma^2_{TFBS}) \\ &= -\sum_n \sum_w \hat{a}_{n,w,1} \left\{ -\frac{1}{2} \left(\frac{s_{n,w} - \mu_{TFBS}}{\sigma_{TFBS}} \right)^2 + \log \frac{1}{\sqrt{2\pi}} - \log \sigma_{TFBS} \right\} - \log \frac{\beta^\alpha}{\Gamma(\alpha)} + 2(\alpha+1) \log \sigma_{TFBS} - \frac{\beta}{\sigma^2_{TFBS}} \end{aligned} \quad \text{(S-15)}$$

Minimizing the upper bound of (S-5) is equivalent to finding σ_{TFBS} where $\frac{df(\sigma_{TFBS})}{d\sigma_{TFBS}} = 0$.

The derivative of (S-15), relative to σ_{TFBS} , is computed as

$$\frac{df(\sigma_{TFBS})}{d\sigma_{TFBS}} = -\frac{1}{\sigma_{TFBS}^3}\sum_n\sum_w \hat{a}_{n,w,1}(s_{n,w}-\mu_{TFBS})^2 + \frac{1}{\sigma_{TFBS}}\sum_n\sum_w \hat{a}_{n,w,1} + 2(\alpha+1)\frac{1}{\sigma_{TFBS}} - 2\frac{\beta}{\sigma_{TFBS}^3} = 0 \quad \text{(S-16)}$$

Finally, we can estimate $(\sigma_{TFBS}^2)*$ as

$$(\sigma_{TFBS}^2)* = \frac{2\beta + \sum_n\sum_w \hat{a}_{n,w,1}(s_{n,w}-\mu_{TFBS})^2}{(2\alpha+2+\sum_n\sum_w \hat{a}_{n,w,1})}. \quad \text{(S-17)}$$

Step 5: Estimate $\lambda*$

Extending (S-5) and excluding items independent on λ , we obtain $f(\lambda)$ as

$$\begin{aligned} f(\lambda) &= -\sum_n\sum_w \hat{a}_{n,w,1}\log P(d_{n,w} \mid b_{n,w}=1,\lambda) \\ &= -\sum_n\sum_w \hat{a}_{n,w,1}\log\left(\frac{1}{2}\exp\left(-\lambda|w|\Delta d\right)\left(1-\exp(-\lambda\Delta d)\right)\right) \end{aligned} \quad \text{(S-18)}$$

Minimizing the upper bound of (S-5) is equivalent to finding λ where $\frac{df(\lambda)}{d\lambda}=0$.

The derivative of (S-18), relative to λ, is computed as follows:

Let $x=\exp(-\lambda\Delta d)$, then we have $f(x) = -\sum_n\sum_w \hat{a}_{n,w,1}\log\left(\frac{1}{2}x^{|w|}(1-x)\right)$.

$$\frac{df(x)}{dx} = -\sum_n\sum_w \hat{a}_{n,w,1}|w|\frac{1}{x} + \sum_n\sum_w \hat{a}_{n,w,1}\frac{1}{1-x} = 0 \quad \text{(S-19)}$$

$$x = \frac{\sum_n\sum_w \hat{a}_{n,w,1}|w|}{\sum_n\sum_w \hat{a}_{n,w,1} + \sum_n\sum_w \hat{a}_{n,w,1}|w|} \quad \text{(S-20)}$$

Finally, we can estimate $\lambda*$ as

$$\lambda* = \frac{1}{\Delta d}\ln\frac{1}{x} = \frac{1}{\Delta d}\ln\left(\frac{\sum_n\sum_w \hat{a}*_{n,w,1} + \sum_n\sum_w \hat{a}*_{n,w,1}|w|}{\sum_n\sum_w \hat{a}*_{n,w,1}|w|}\right) \quad \text{(S-21)}$$

S6. Parameter settings for competing peak calling methods

The most recent versions of tools used for comparison in this study are downloaded from the website and the settings or commands are listed as follows to run each tool properly.

1. MACS (version 1.4.2): we run MACS with default settings but varying *p*-value threshold from 1e-300 to 0.5 to calculate its precision/recall performance curve, of which the best performance in terms of F-measure is shown in Table 1 in the main text. The reported *p*-value is in its log-transformed (–10*log10) format so the range is [3, 3000]. We divide the whole range into 1000 segments with 3 as the incensement step, and calculate precision-recall values under each threshold. Default *p*-value threshold 1e-5 (50 in -10*log10 format) is also used for performance comparison (as reported in Table S1 and S2 in the next section).

2. PeakSeq (version 1.31): we run PeakSeq with default settings but varying Q-value from 0 to 1 to calculate its precision/recall performance curve, of which the best performance in terms of F-measure is shown in Table 1 in the main text. The range of Q-value is (0, 1]. We divide this range into 1000 segments with 0.001 as the incensement step, and calculate precision-recall values under each threshold. Default Q-value threshold 0.05 is also used for performance comparison (as reported in Table S1 and S2).

3. BCP (version 1.1): we run BCP with default settings but varying *p*-value threshold from 1e-16 to 0.5 to calculate its precision/recall performance curve, of which the best performance in terms of F-measure is shown in Table 1 in the main text. We transform this *p*-value into its log-transformed (-10log10) format so the overall range becomes [3, 160]. We divide this range into 1000 segments with 0.16 as the incensement step, and further calculate precision-recall values under each threshold. Default *p*-value threshold 1e-8 (80 in -10log10 format) is also used for performance comparison (as reported in Table S1 and S2).

4. Dfilter (version 1.5): we run Dfilter with recommended options for transcription factor ChIP-seq analysis: -nonzero -bs=50 -ks=20 -refine -std=2. The *p*-value threshold is varied from 1e-200 to 0.5 so as to present its precision/recall performance curve, of which the best performance in terms of F-measure is shown in Table 1 in the main text. The reported *p*-value is in its log-transformed (–log10) format such that its overall range is [0.3, 200]. We divide the range into 1000 segments with 0.2 as the incensement step, and calculate precision-recall values under each threshold. Default *p*-value threshold 1e-6 (6 in –log10 format) is also used for performance comparison (as reported in Table S1 and S2).

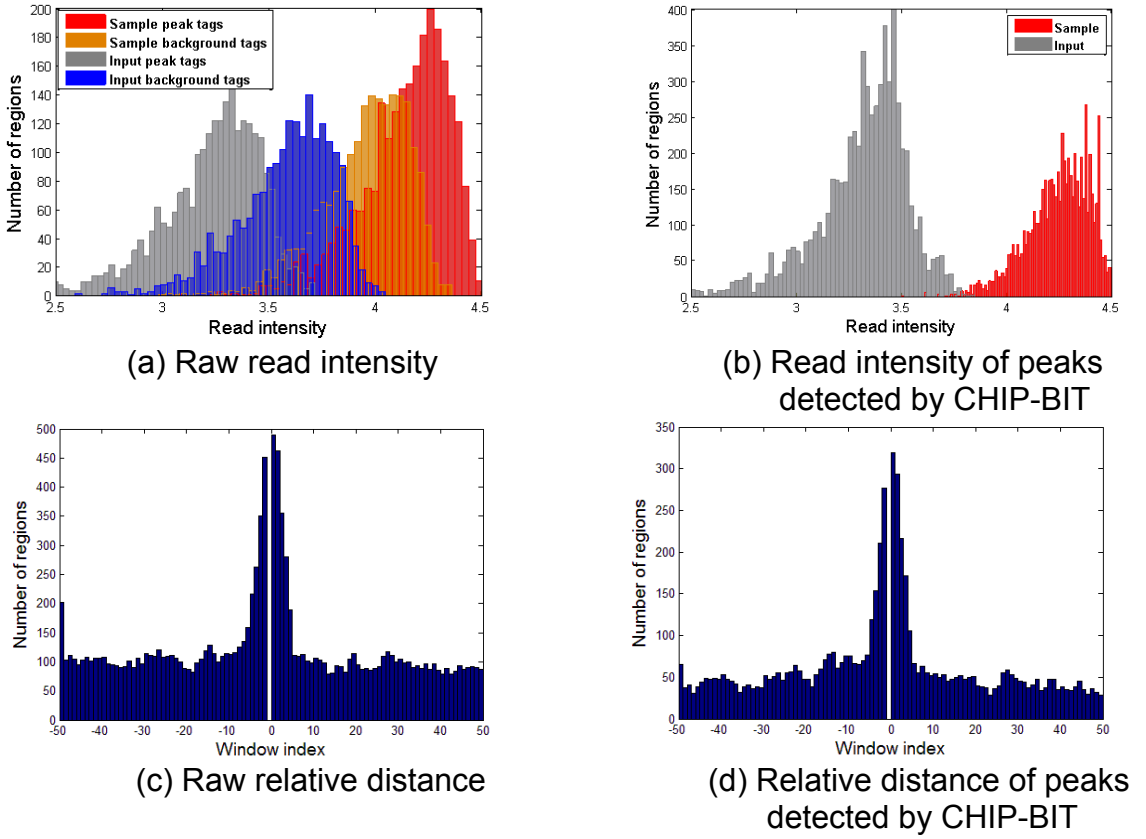
5. MOSAiCS (version 1.2.2): we run MOSAiCS with default settings; the performance of MOSAiCS is shown is Table S1 and S2. Based on the reported average log ratio, we can further rank all peaks and calculate the overall precision/recall curve by varying the cut-off threshold, of which the best performance is shown in Table 1 in the main text. In detail, we take the maximum value and minimum value of the average log ratio to determine its overall range. Then, we divide the overall range into 1000 segments and calculate precision-recall values under each threshold.

[Note that simulated peaks are distant from each other at least 500 bps to avoid the peak overlap problem when a low threshold is used.]

S7. Simulation studies

S7.1. Simulation data generation

Case 1: We spread peaks at promoter regions according to an exponential distribution, as shown in Fig. S10(c). Read intensity distributions of true peaks and background regions are close in sample data but quite different in input data, as shown in Fig. S10(a). This is consistent to real data since background regions have highly correlated read enrichment in both sample and input data while true peaks are condition-specific and enriched in sample data only.



(a) Raw read intensity

(b) Read intensity of peaks detected by CHIP-BIT

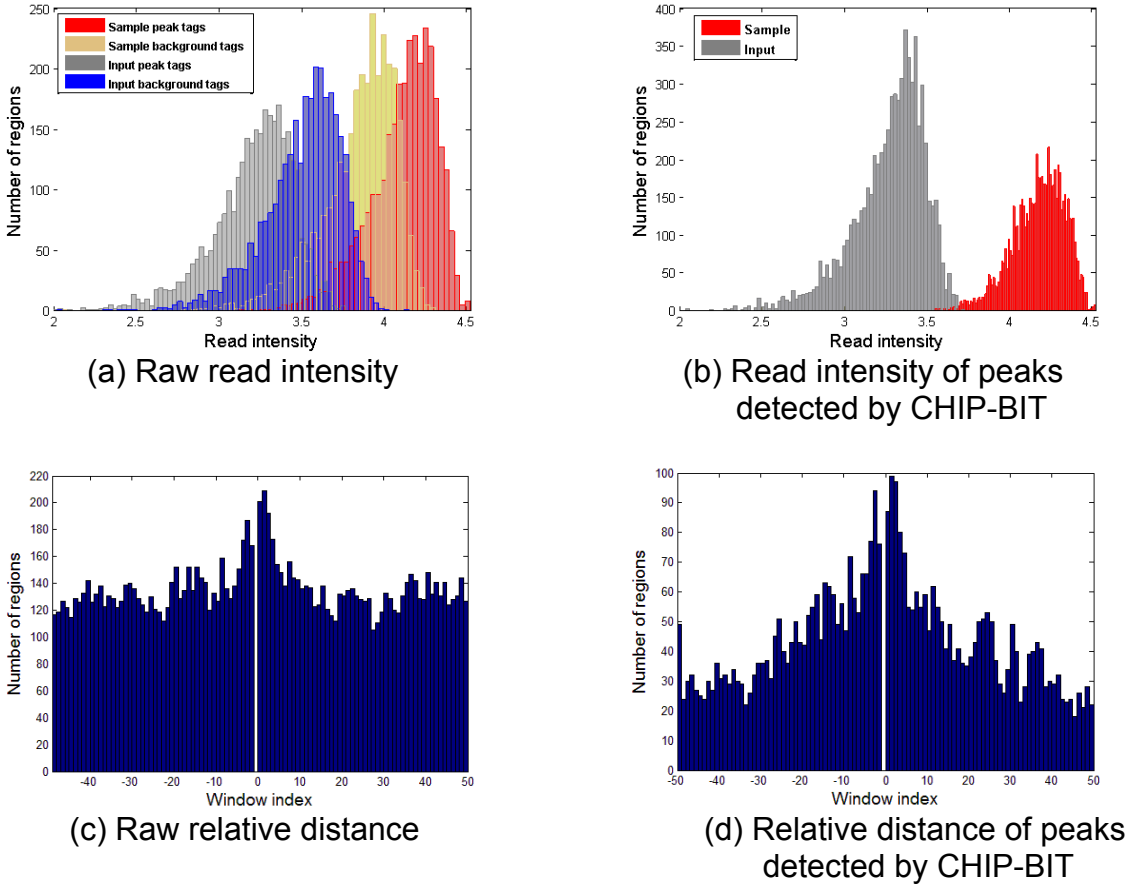
(c) Raw relative distance

(d) Relative distance of peaks detected by CHIP-BIT

Fig. S10. Simulation results of Case 1.

In Fig. S10(b) and (d), CHIP-BIT successfully detected peaks by estimating their read intensity distribution consistently to the distribution of 'true' peaks in the raw data. In addition, the distances of detected peaks relative to TSS are exponentially distributed as well.

Case 2: We spread peaks at promoter regions more evenly, as shown in Fig. S11(c). In addition, in this case, we lowered the fold change of peaks by increasing its input read intensity. In Fig. S11(a), the distribution of 'gray' signals is closer to the distribution of 'blue' signal comparing to the situations in Fig. S10(a).



(a) Raw read intensity

(b) Read intensity of peaks detected by CHIP-BIT

(c) Raw relative distance

(d) Relative distance of peaks detected by CHIP-BIT

Fig. S11. Simulation results of Case 2.

In Fig. S11(b) and (d), CHIP-BIT successfully **d**etected peaks by estimating their read intensity distribution consistently to the distribution of 'true' peaks in the raw data. The distances of detected peaks relative to TSS are more evenly distributed.

S7.2. Peak detection using default settings for each competing method

Table S2. Precision/recall on peak detection with default settings for Case 1

	Case 1	CHIP-BIT	PeakSeq	MACS	BCP	Dfilter	MOSAiCS
All peaks	Precision	0.9057	0.8526	0.8078	0.8458	0.7492	0.9527
	Recall	0.8784	0.7883	0.8084	0.7328	0.8076	0.6764
	F-measure	0.8918	0.8192	0.8081	0.7853	0.7773	0.7911
Weak peaks	Precision	0.8468	0.7522	0.8555	0.8736	0.7298	0.9173
	Recall	0.8061	0.6523	0.6749	0.5499	0.7526	0.5221
	F-measure	0.8260	0.7000	0.7546	0.6750	0.7410	0.6655

Table S3. Precision/recall on peak detection with default settings for Case 2

	Case 1	CHIP-BIT	PeakSeq	MACS	BCP	Dfilter	MOSAiCS
All peaks	Precision	0.9028	0.8906	0.8432	0.8756	0.7373	0.8848
	Recall	0.8639	0.7643	0.7803	0.7132	0.7681	0.7506
	F-measure	0.8830	0.8241	0.8105	0.7861	0.7524	0.8122
Weak peaks	Precision	0.8345	0.8652	0.8461	0.8697	0.8807	0.8373
	Recall	0.8058	0.5749	0.6280	0.5108	0.3402	0.6214
	F-measure	0.8199	0.6908	0.7209	0.6436	0.4908	0.7134

Table S4. Precision/recall on boundary detection for Case 1 (true positive peaks only)

Case 1	CHIP-BIT	PeakSeq	MACS	BCP	Dfilter	MOSAiCS
Precision	0.9098	0.9104	0.7538	0.7351	0.7332	0.7746
Recall	0.7703	0.8111	0.9511	0.8049	0.9882	0.9997

Table S5. Precision/recall on boundary detection for Case 2 (true positive peaks only)

	CHIP-BIT	PeakSeq	MACS	BCP	Dfilter	MOSAiCS
Precision	0.9086	0.7705	0.8661	0.7171	0.7524	0.7677
Recall	0.7614	0.8857	0.9284	0.6515	0.9644	0.9861

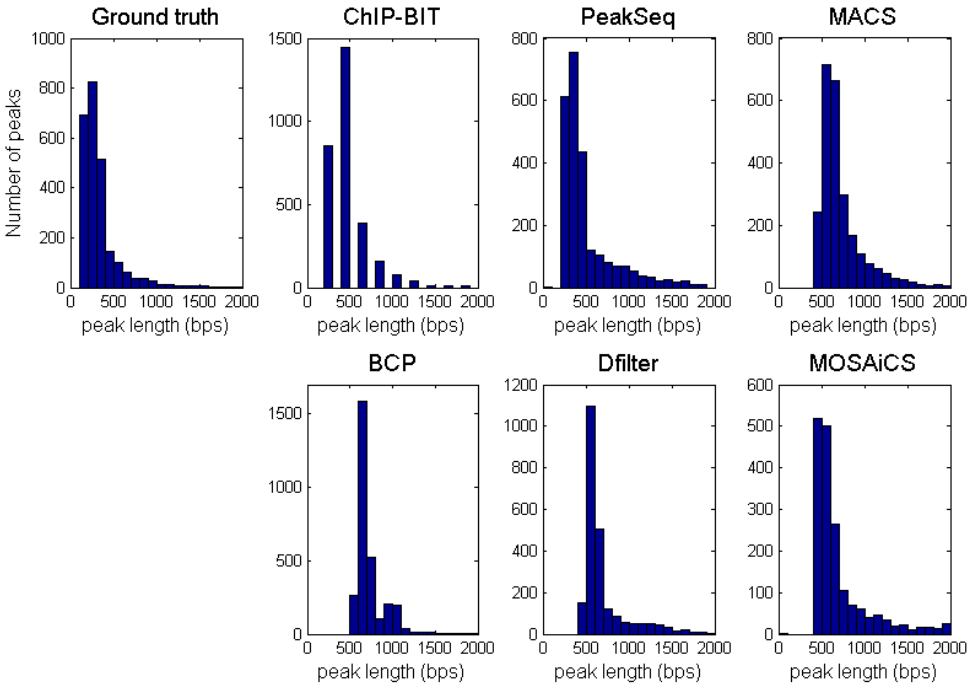


Fig. S12. Peak length distribution of peaks detected by each competing method for Case 1.

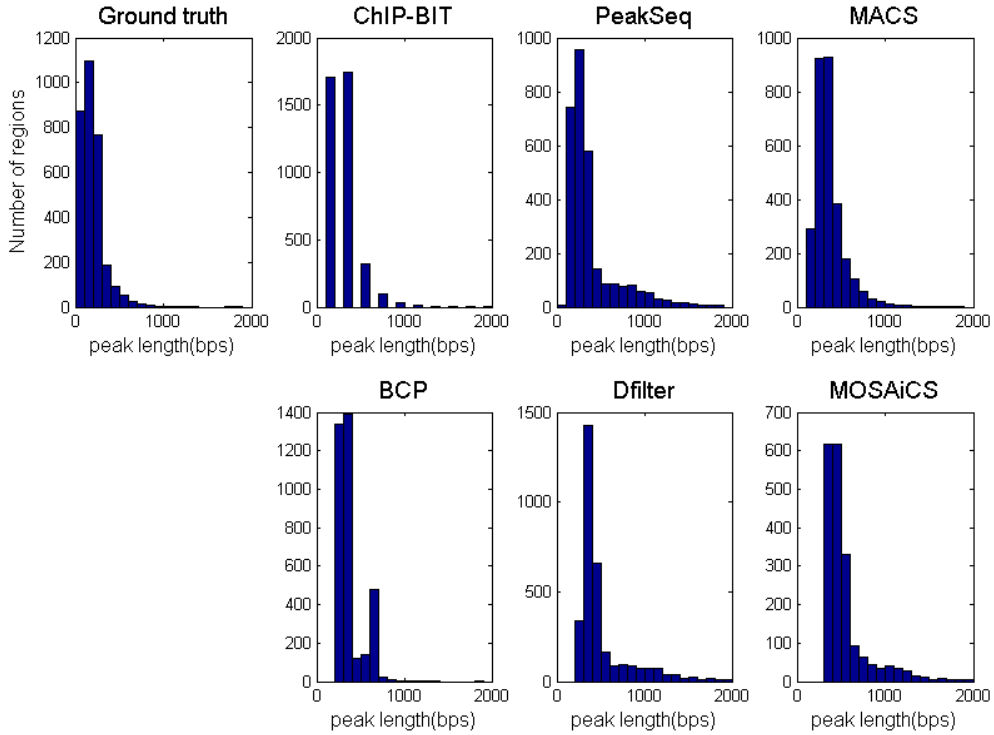


Fig. S13. Peak length distribution of peaks detected by each competing method for Case 2.

S7.3. Target gene prediction

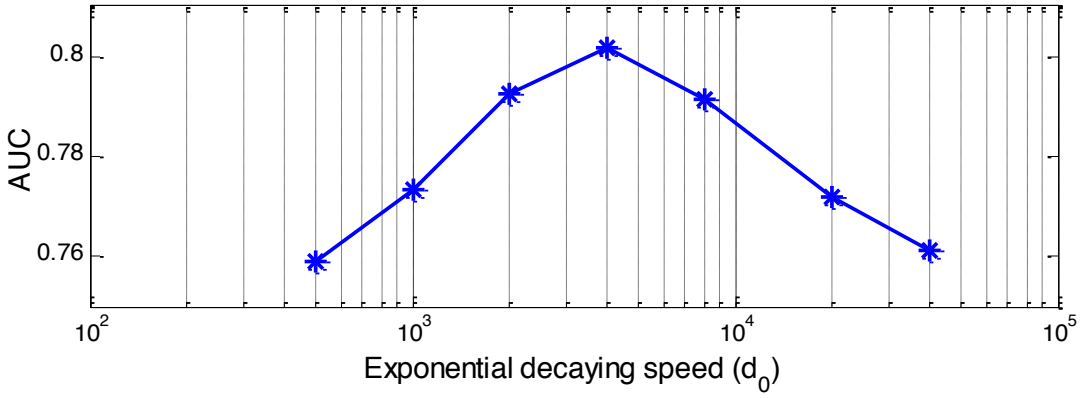
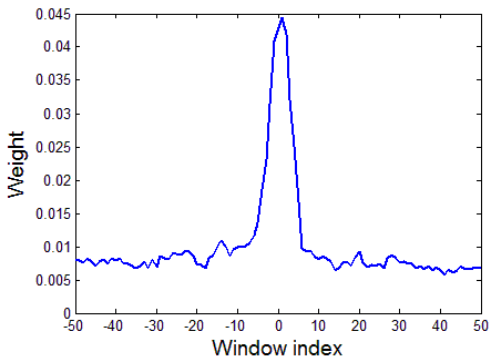
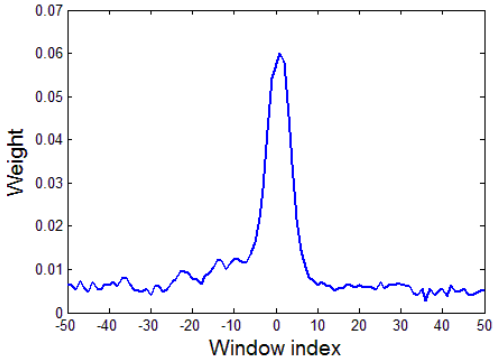


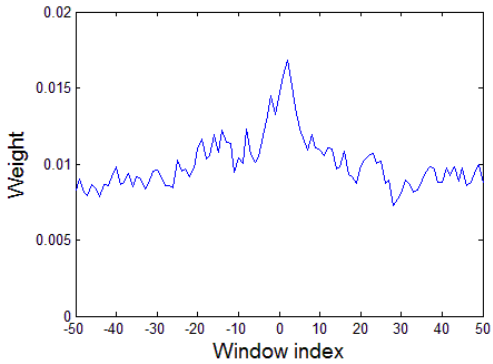
Fig. S14. Gene prediction performance of the method proposed by Ouyang *et al.* (9) with different decaying speeds (simulation Case 1 with true promoter region ±5k).



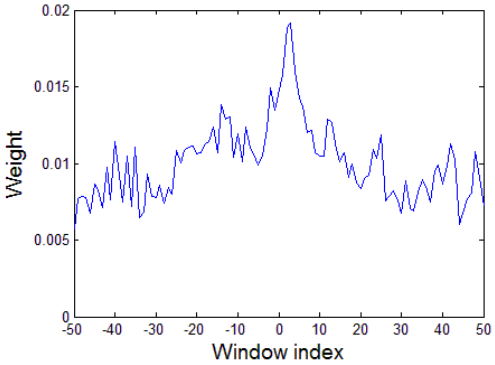
(a) Weight vector learned by TIP for Case 1



(b) Weight vector learned by Improved TIP for Case 1



(c) Weight vector learned by TIP for Case 2



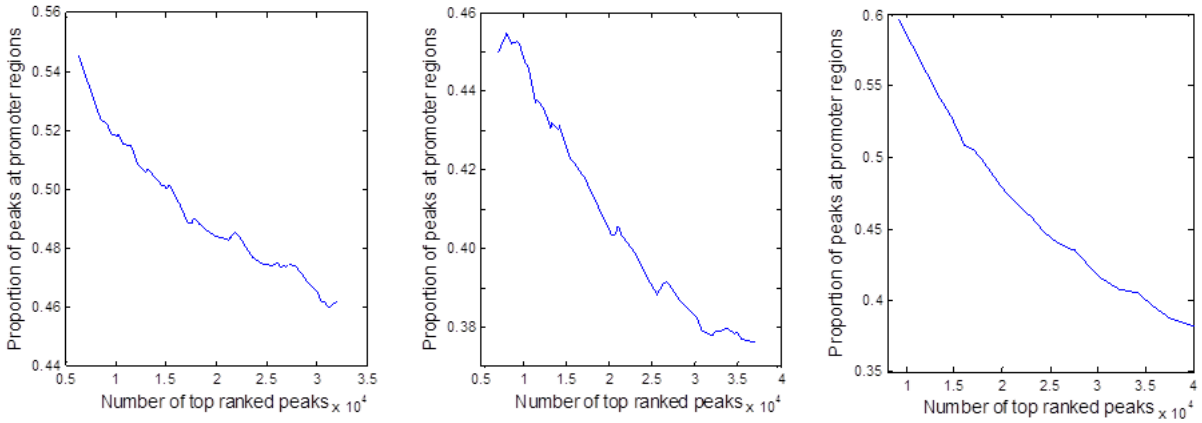
(d) Weight vector learned by Improved TIP for Case 2

Fig. S15. Weight vectors of TIP and Improved TIP for target gene prediction.

S8. Performance comparison using real ChIP-seq data

S8.1. Identified TFBSs and validated benchmark regions

In order to compare with the original results resented by Rye *et al*. (10), we directly download their processed bed format ChIP-seq data of MAX, NRSF and SRF as well as specific benchmarks from http://tare.medisin.ntnu.no/chipseqbenchmark/. In this study, we focus on binding sites or peaks that lay in gene promoter regions. Therefore, it is necessary to check the proportion of peaks within gene promoter regions (f_{Promoter}) among peaks detected from the whole genome. For each TF, whole genome peaks are detected by PeakSeq with default settings and then sorted according to the Q-value. A f_{Promoter} curve obtained using different Q-value thresholds (0~0.05) is shown in Fig. S16.



(a) MAX (b) NRSF (c) SRF

Fig. S16. Proportion of peaks at gene promoter regions.

In another independent ChIP-seq study (11) on over 100 TFs under K562 cell line, it is reported that ~40% of identified peaks are proximal (within ±2.5 kbps) to TSSs, where f_{Promoter} = 40%. For these three representative ChIP-seq data sets, we use a larger promoter region as ±10 kbps. Thus, f_{Promoter} increases to ~50%.

We filter all benchmarks with gene promoter regions and the remaining positive or negative benchmarks are summarized in Table S6.

Table S6. Benchmarks of MAX, NRSF and SRF

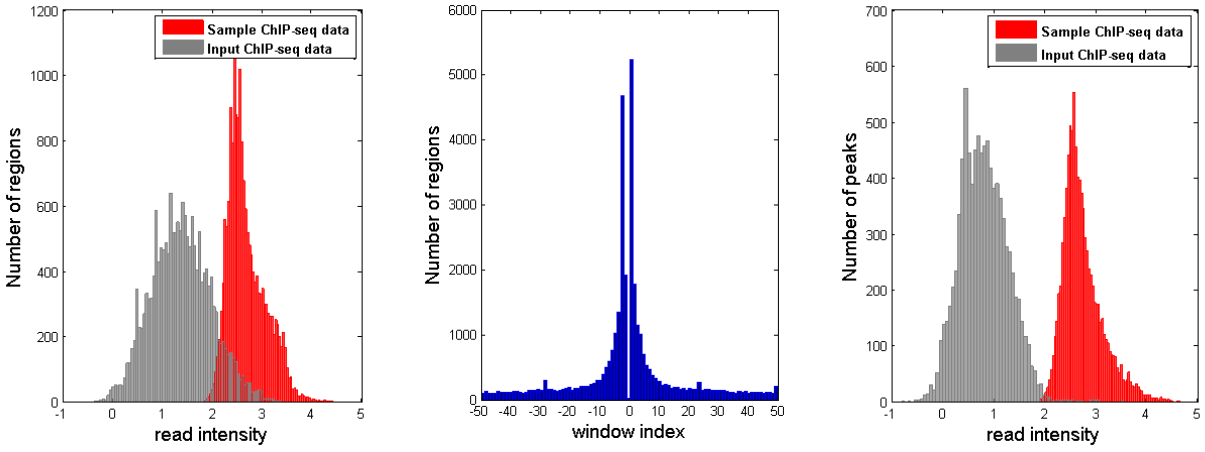
	Whole genome			Promoter region		
	Positive	ambiguous	Negative	Positive	ambiguous	Negative
MAX	163	51	200	119 (73%)	22	68
NRSF	127	30	322	57 (45%)	14	132
SRF	124	16	311	86 (70%)	8	112

We call peaks using all methods respectively with their default settings and filter reported peaks using gene promoter regions. The number of peaks that lay in gene promoter regions for each TF and each method is shown in Table S7.

Table S7. No. of promoter region bound peaks of MAX, NRSF and SRF

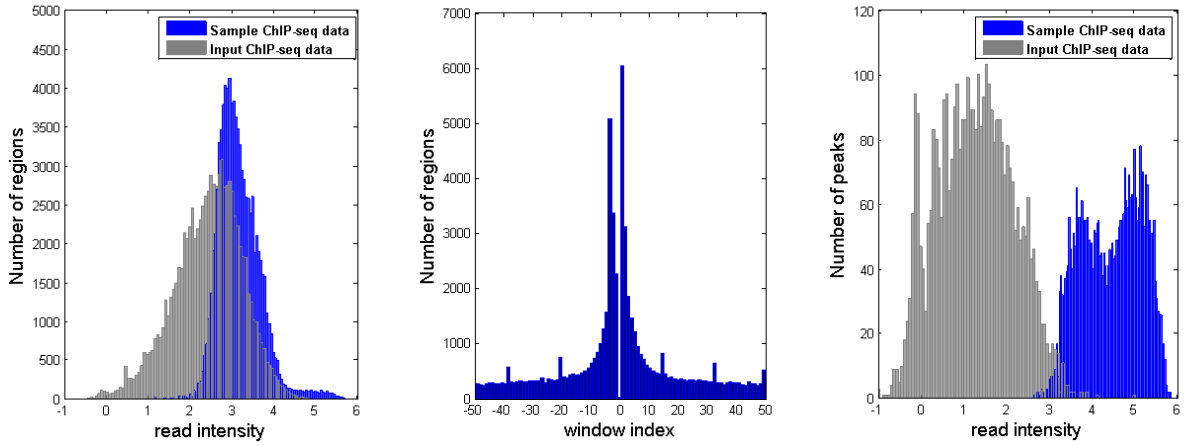
	ChIP-BIT	PeakSeq	BCP	MOSAiCS	MACS	Dfilter
MAX	9,281	10,704	5,521	8,645	11,926	7,753
NRSF	2,697	7,117	2,566	2,359	1,875	2,844
SRF	1,497	1,037	1,003	1,971	2,060	1,284

For ChIP-BIT, we present the histograms of read intensities and relative distance to TSS of regions before and after peak calling in Fig. S17 - S19 for MAX, NRSF and SRF, respectively.



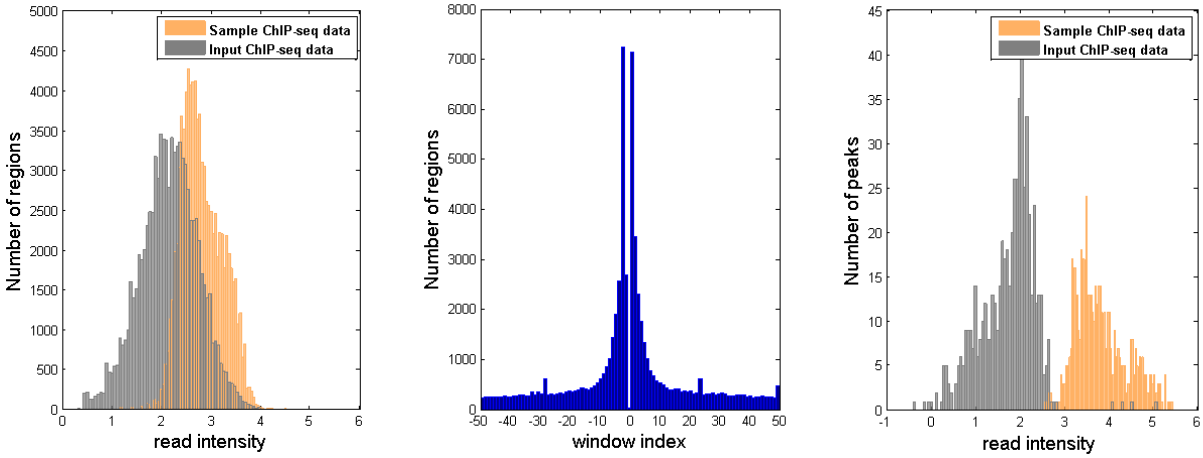
(a) Raw read intensity (b) Distance to TSS (c) ChIP-BIT

Fig. S17. Read intensities and binding locations of ChIP-BIT detected MAX peaks.



(a) Raw (b) Distance to TSS (c) ChIP-BIT

Fig. S18. Read intensities and binding locations of ChIP-BIT detected NRSF peaks.

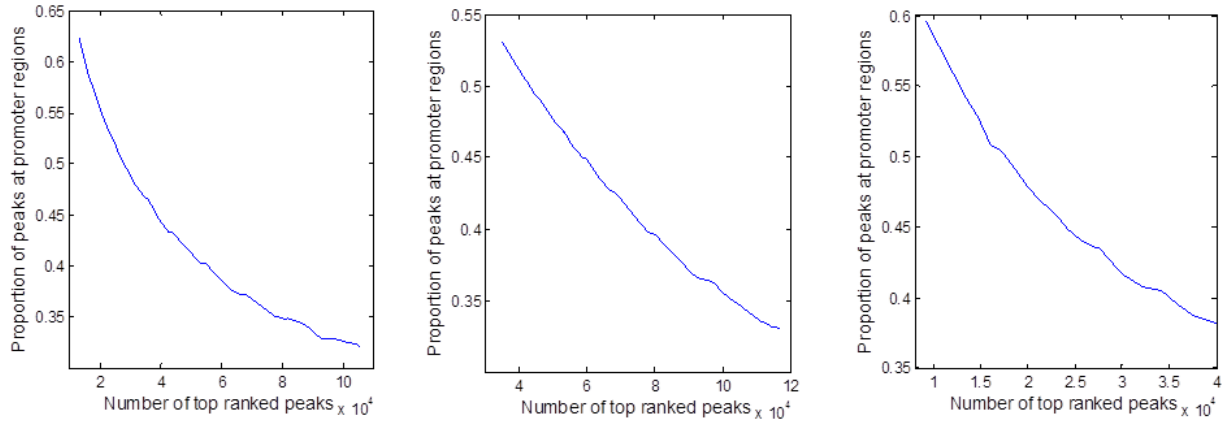


(a) Raw (b) Distance to TSS (c) ChIP-BIT

Fig. S19. Read intensities and binding locations of ChIP-BIT detected SRF peaks.

S8.2. Identified target genes and RNA-seq profiling

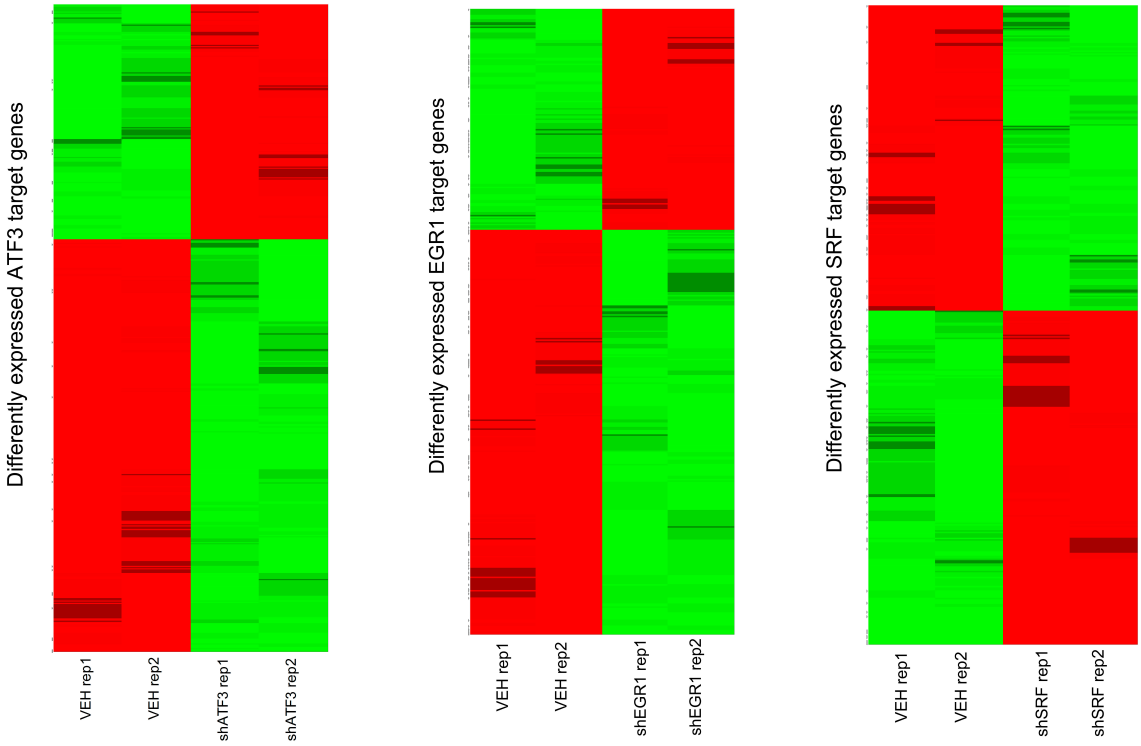
To compare target gene prediction performance of different methods, we applied ChIP-BIT, MACS, PeakSeq, and TIP to ATF3, EGR1 and SRF ChIP-seq data. ChIP-seq data is downloaded from ENCODE project under K562 cell line http://genome.ucsc.edu/ENCODE/. Since MACS and PeakSeq only report peak locations, we use GREAT to do gene annotation with upstream/downstream 10k bps, the same promoter region setting as ChIP-BIT or TIP. Similar to Fig. S16, we present f_{Promoter} (proportion of peaks that lay in promoter regions) for each TF in Fig. S20.



(a) ATF3 (b) EGR1 (c) SRF

Fig. S20. Proportion of peaks at gene promoter regions.

Then, we use the matched RNA-seq data generated before or after specific TF knockdown to validate genes identified by each method on each data set. The RNA-seq data is downloaded from GEO data base under access number GSE33816. For each TF, there are two replicates under control or treatment conditions (Vehicle vs. shRNA). We apply Tophat (2.0.14) to the fastq files of each RNA-seq sample and use Cufflinks (2.2.1) to estimate the abundance of each transcript (FPKM) across all samples. Finally, we use t-test to identify differentially expressed genes by assuming that at least one transcript of each gene has a significant expression change with p-value<0.05 after shRNA introduction. In total we identified 2,810 genes for ATF3, 3,356 genes for EGR1 and 3,401 genes for SRF, whose expression patterns are shown in Fig. S21.



(a) ATF3 (b) EGR1 (c) SRF

Fig. S21. Heat map of differentially expressed target genes after knocking down each specific TF.

We map differentially expressed gene list to the target genes identified from ChIP-seq analysis using individual method. Direct target genes showing differential expression for each method are shown in Table S8 – S10 for ATF3, EGR1 and SRF, respectively.

Table S8. Differentially expressed target genes of ATF3

Methods	**ChIP-BIT**	MACS	PeakSeq	TIP
Target genes of ChIP-seq	**2,443**	2,963	8,680	4,255
Differentially expressed target genes	**285 (11.67%)**	342 (11.54%)	1,005 (11.58%)	401 (9.42%)
Average overlap with other methods	**0.5054**	0.4883	0.2262	0.1347

Table S9. Differentially expressed target genes of EGR1

Methods	**ChIP-BIT**	MACS	PeakSeq	TIP
Target genes of ChIP-seq	**7,190**	9,454	11,707	5,600
Differentially expressed target genes	**1,202 (16.72%)**	1,547 (16.36%)	1,866 (15.94%)	762 (13.61%)
Average overlap with other methods	**0.7371**	0.6596	0.5681	0.5402

Table S10. Differentially expressed target genes of SRF

Methods	**ChIP-BIT**	MACS	PeakSeq	TIP
Target genes of ChIP-seq	**3,582**	6,620	4,243	3,747
Differentially expressed target genes	**568 (15.86%)**	1,047 (15.82%)	686 (16.17%)	429 (11.45%)
Average overlap with other methods	**0.5088**	0.3187	0.5165	0.3131

S8.3. TF association under K562 cell line using peaks identified by ChIP-BIT

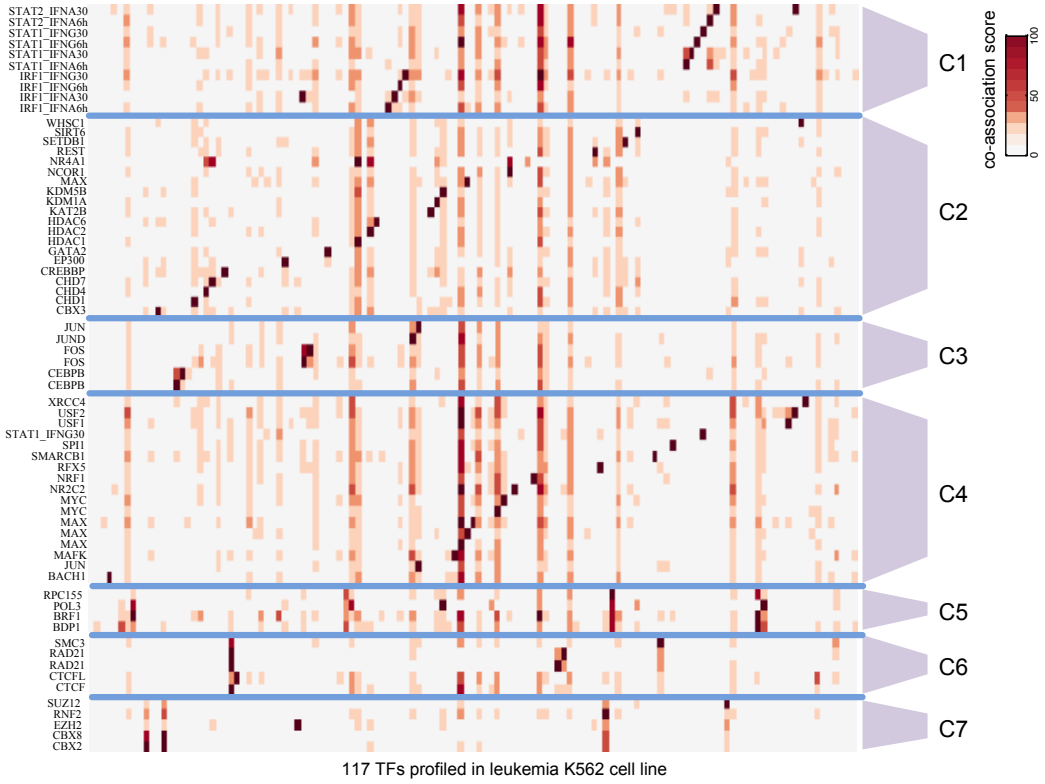
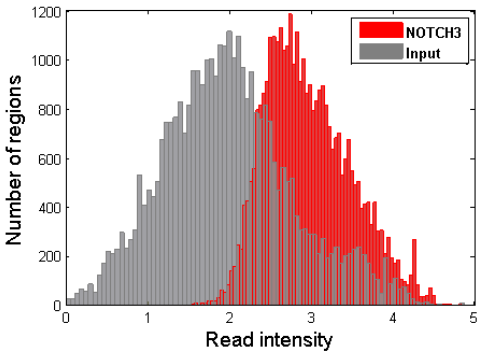


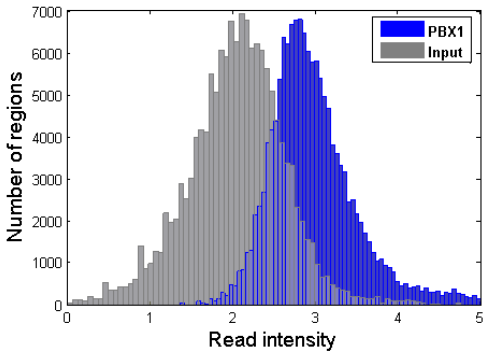
Fig. S22. Identified TF co-association pattern in K562 cell line by using peaks detected by ChIP-BIT. Note that the color represents the co-association score calculated according to (11).

Cluster **C1** represents a known co-operation between STAT1/2 and IRF1 in IFNα/γ stimulation. **C2** highlights the interaction between chromatin remodelers including CHD complex and HDAC complex. In Cluster **C3** and **C4**, two stable co-associations like JUN-FOS and MYC-MAX are respectively identified. **C5** is a TF cluster from POL3 complex, including RPC155, POL3, BRF1 and BDP1. Cluster **C6** consists of CTCF, RAD21, and SMC3, which are known to form large complexes regulating chromatin structure. **C7** is a very unique co-association with five TFs from Polycomb group.

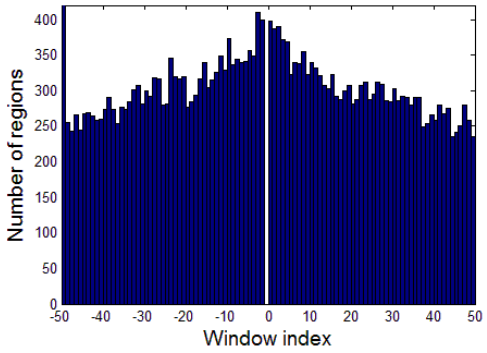
S9. NOTCH3 and PBX1 ChIP-seq data analysis



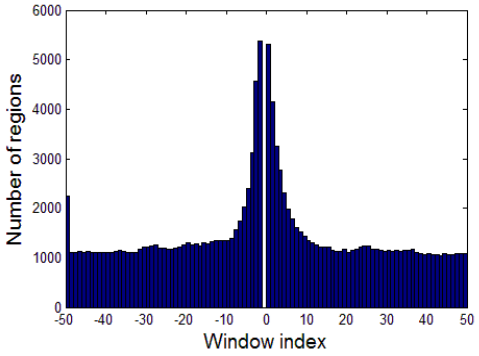
(a) Read intensity at NOTCH3 candidate regions



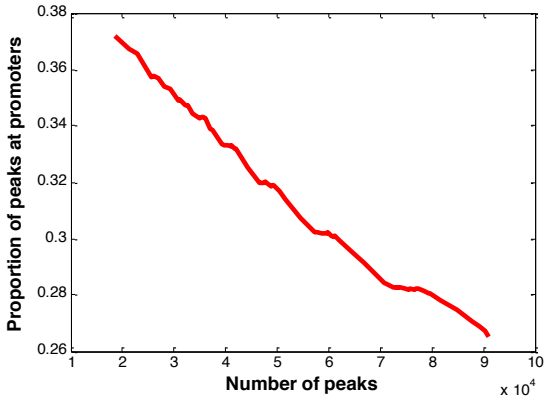
(b) Read intensity at PBX1 candidate regions



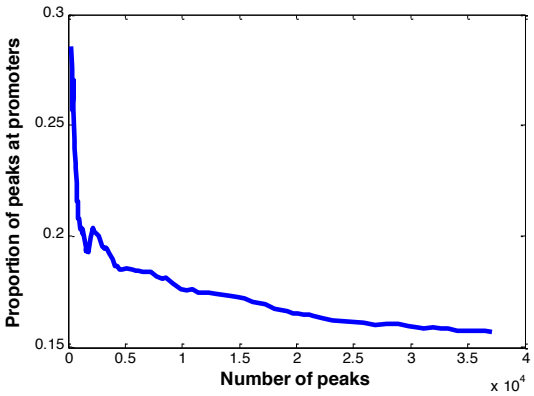
(c) Relative distance of NOTCH3 candidate regions



(d) Relative distance of PBX1 candidate regions



(e) Proportion of NOTCH3 peaks at gene promoter regions



(f) Proportion of PBX1 peaks at gene promoter regions

Fig. S23. Raw distributions of NOTCH3 and PBX1 ChIP-seq data

Table S11. Read intensity distribution of peaks detected using different thresholds

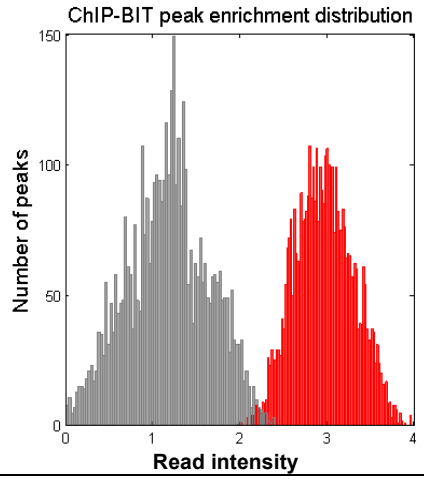
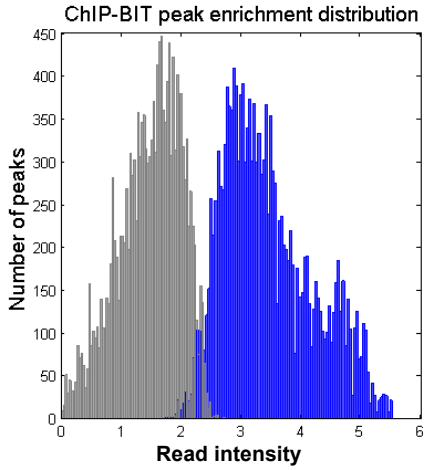
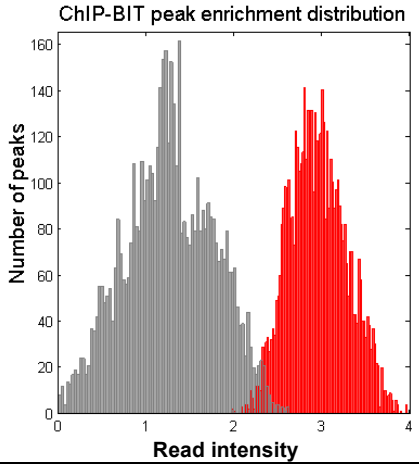
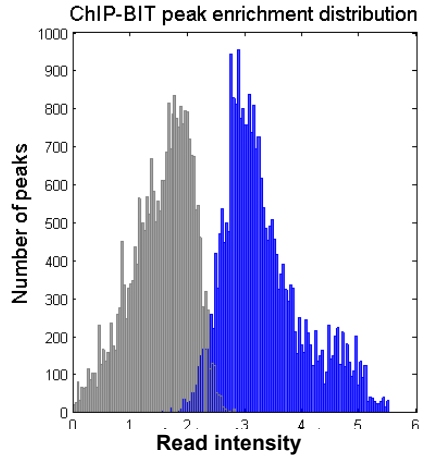
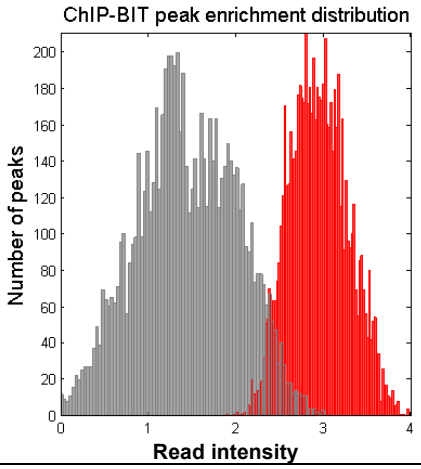
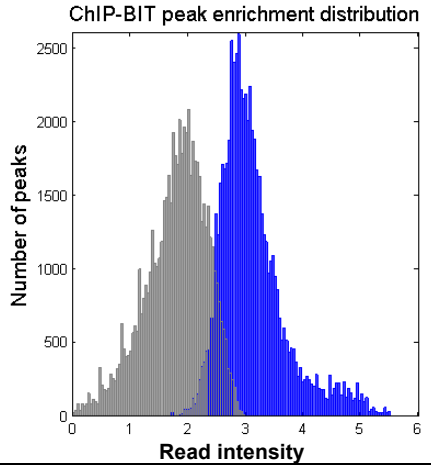
Threshold	NOTCH3	PBX1
0.9		
0.8		
0.7		

Table S12. Peaks detected by each method

TF	Method	Number of Peaks	Mean length(bps)	Standard deviation (bps)
NOTCH3	**ChIP-BIT**	**3,288**	**530**	**363**
	PeakSeq	15,057	603	396
	MACS	10,076	1,731	685.3
PBX1	**ChIP-BIT**	**6,022**	**443**	**304**
	PeakSeq	18,581	703	622
	MACS	23,013	1243	464

For TIP, we sort all genes according to their z-score and select a proper number of targets from the top list to make it comparative to the number of genes reported by ChIP-BIT.

Table S13. Target genes detected by competing methods

TF	Method	Number of Genes	Average overlap with other methods
NOTCH3	**ChIP-BIT**	**2,871**	**0.38**
	PeakSeq	2,657	0.24
	MACS	3,001	0.22
	TIP	2,749	0.22
PBX1	**ChIP-BIT**	**5,280**	**0.61**
	PeakSeq	6,546	0.48
	MACS	7,333	0.54
	TIP	5,564	0.48

S10. Functional annotation of common target genes of PBX1 and NOTCH3

To identify common target genes, first of all, in each data set, we selected target genes which contain at least one binding site with probability over 0.8. In total, there are about 1936 common target genes. Secondly, we calculate the posterior probability for each gene according to Eq. (19). The distribution of posterior probabilities for common target genes is shown in Fig. S24. Finally, by setting the cut-off threshold as 0.95, we identified 621 common targets.

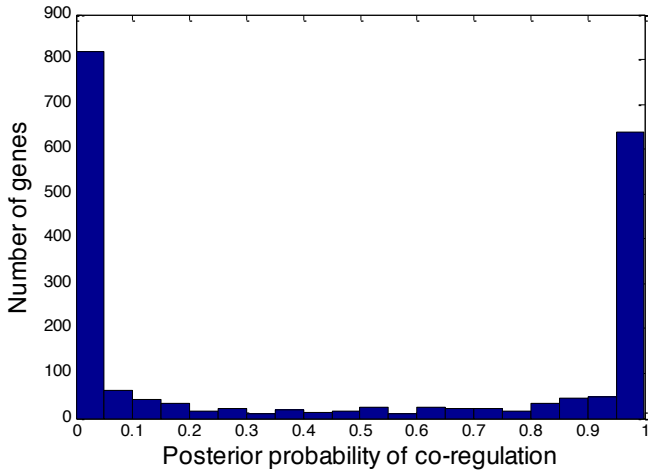


Fig. S24. Distribution of posterior probabilities for common target genes identified from NOTCH3 and PBX1 ChIP-seq data.

Table S14. Target gene list of NOTCH3, PBX1 or both detected by ChIP-BIT

Table S14 can be found in "ChIP-BIT-Suppl-TableS14.xlsx".

We further identify common target genes of NOTCH3 and PBX1 using other method by taking the intersection of genes regulated by individual TF, as shown in Table S15.

Table S15. Numbers of common target genes detected by competing methods

	ChIP-BIT	PeakSeq	MACS	TIP
NOTCH3	**2,871**	2,657	3,001	2,749
PBX1	**5,280**	6,546	7,333	5,564
Common	621	869	995	1535

S10.1 Notch signaling pathway

Notch signaling pathway data base links:

KEGG Notch signaling pathway
(http://www.genome.jp/dbget-bin/www_bget?pathway+hsa04330)

NCI/Nature Notch signaling pathway
(http://pid.nci.nih.gov/search/pathway_landing.shtml?pathway_id=200015&source=NATURE&what=graphic&jpg=on)

QIAGEN Notch signaling pathway
(http://www.qiagen.com/us/products/genes%20and%20pathways/complete%20biology%20list/notch%20signaling/rt2-profiler-pcr-arrays?catno=PAHS-059Z#geneglobe)

Table S16. Notch signaling pathway enriched NOTCH3 target genes

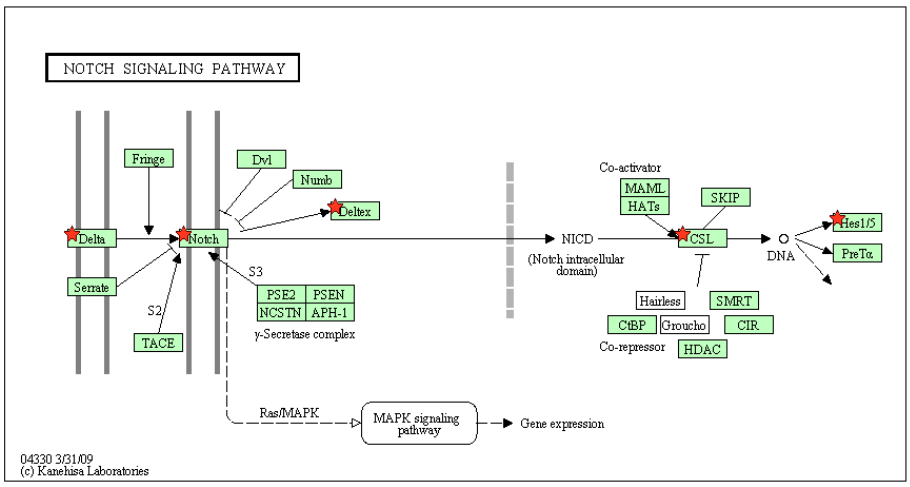
	ChIP-BIT	PeakSeq	MACS	TIP
NOTCH3 signaling pathway	22	14	11	16
p-value	0.015	>0.05	>0.05	>0.05

ChIP-BIT identified 22 Notch signaling pathway enriched NOTCH3 target genes in Table S15 include ARNT, BLOC1S1, CNTN1, CNTN6, DLK1, DLL4, DTX3L, DTX4, FIGF, GLI1, HDAC2, HES1, HEY1, LEF1, NCOA1, NCOR2, NFKB1, NOTCH2, NOTCH4, RBPJ, SH2D1A, and TLE1.

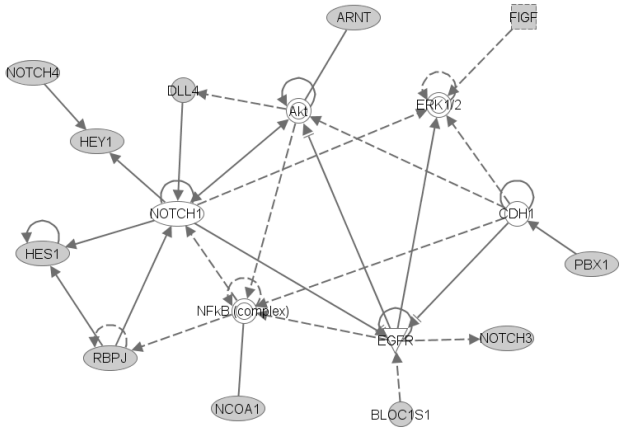
Table 17 Notch signaling pathway enriched common target genes

	ChIP-BIT	PeakSeq	MACS	TIP
NOTCH3 signaling pathway	11	4	0	8
p-value	0.0016	>0.1	-	>0.1

ChIP-BIT identified 11 Notch signaling pathway enriched common target genes in Table S16 include ARNT, BLOC1S1, CNTN1, DLL4, DTX4, FIGF, HES1, HEY1, NCOA1, NOTCH4, and RBPJ.



(a)



(b)

Fig. S25. Enriched genes on (a) KEGG's Notch signaling pathway and (b) Ingenuity Pathway Analysis (IPA)-defined Notch signaling network.

S10.2. Wnt signaling pathway

Wnt signaling pathway database links:

KEGG Wnt signaling pathway
(http://www.genome.jp/dbget-bin/www_bget?hsa04310)

NCI/Nature Wnt signaling pathway
(http://pid.nci.nih.gov/search/pathway_landing.shtml?pathway_id=200077&source=NATURE&what=graphic&jpg=on)

QIAGEN Wnt signaling pathway
(http://www.qiagen.com/us/products/genes%20and%20pathways/complete%20biology%20list/wnt%20signaling/rt2-profiler-pcr-arrays?catno=PAHS-043Z#geneglobe)

Table S18. Wnt signalling enriched common target genes

	ChIP-BIT	PeakSeq	MACS	TIP
Wnt signaling pathway	11	3	5	7
p-value	0.047	>0.1	>0.1	>0.1

ChIP-BIT identified 11 Wnt signaling pathway enriched common target genes in Table S17 include CEBPD, IL6, KLF5, PIAS1, PRICKLE1, SMAD4, SOX2, SOX9, WNT2, WNT2B, and WNT4.

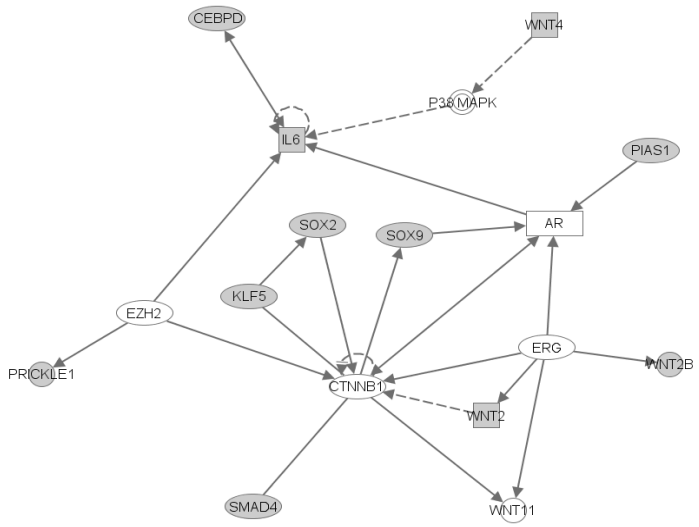


Fig. S26. Wnt signaling pathway defined by IPA network analysis.

S11. Differentially expressed genes

In the GSI or siRNA TF knockdown study, under each condition, we only collected two duplicates of gene expression data. For the pairwise identification of differentially expressed genes, with so few samples, significance test like t-test does not work well. Therefore, in this study, we use the fold change (FC) information and select those genes if they meet the following four conditions simultaneously:

(1) |Sample1 - control1|>FC threshold; (2) |Sample1 – control2|>FC threshold;
(3) |Sample2 - control1|>FC threshold; (4) |Sample2 – control2|>FC threshold.

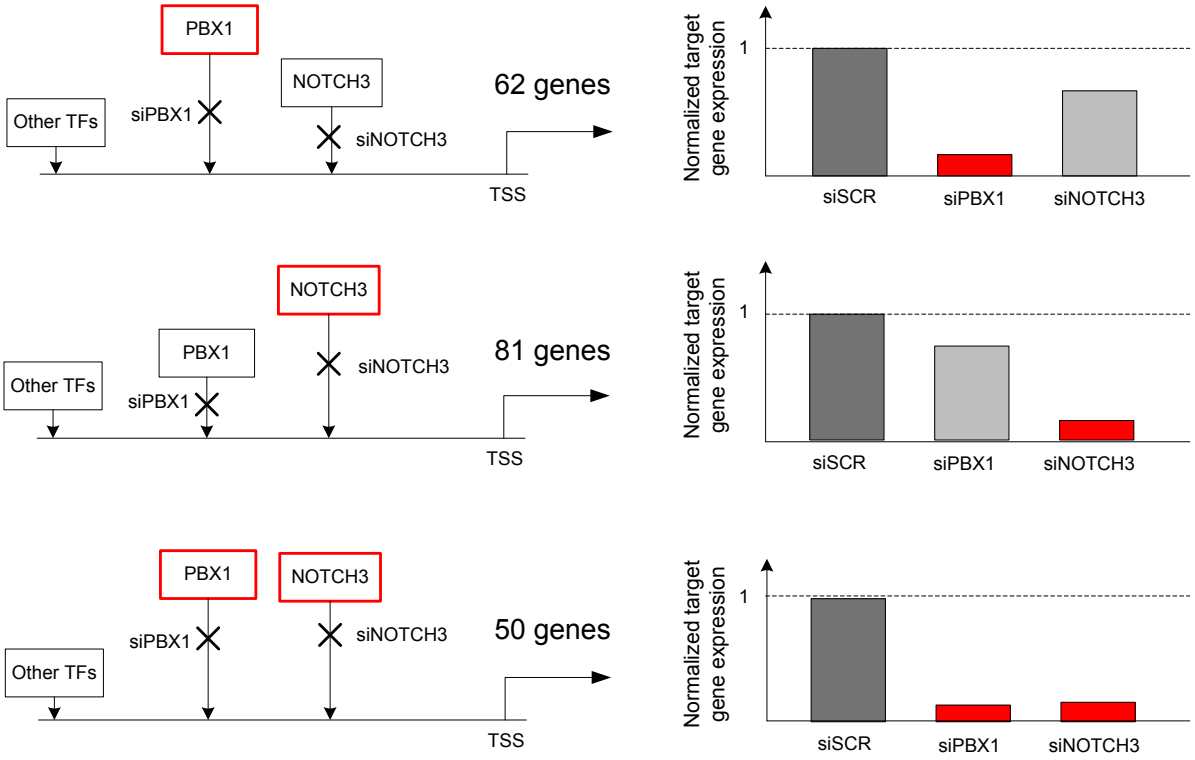


Fig. S27. Relationship of TF knockdown and target gene expression. The major TF (or TFs) is (or are) labeled in ‘red’.

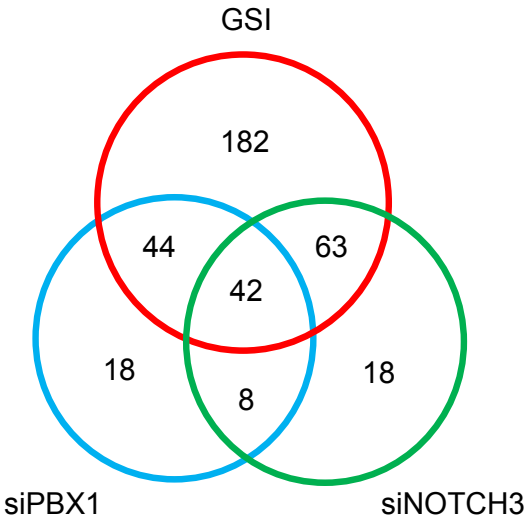


Fig. S28. Overlap of differentially expressed genes under GSI, siNOTCH3 and siPBX1.

Table S19. Differentially expressed gene lists under GSI, siNOTCH3 and siPBX1

Table S19 can be found in "ChIP-BIT-Suppl-TableS19.xlsx".

S12. Glossary of specific terms and variables used in the text and supplementary material

Cluster:	*A set of overlapped 200 bps long fragments extended from read tags*
Segment:	*Non-overlapping 20k bps long genome used for candidate region searching*
l:	*Length of segment, 20k*
Mappability:	*Number of uniquely mappable nucleotides within a segment*
f:	*Fraction of uniquely mappable nucleotides within a segment (mappability/l)*
N:	*Number of fragments in a segment*
Candidate region:	*A significantly enriched region compared to a simulated null background distribution in a segment*
ρ:	*Quantile threshold used to select low read count regions*
N_{region}:	*Number of fragments (read count) overlapping with a candidate region*
C_1:	*Read coverage, number of fragments overlapping with a single nucleotide*

Bin:	*Non-overlapping 50 bps long genome of a candidate region*
C_{50}:	*Average read coverage of a bin,* $\sum_{i=1}^{50} C_{1,i} / 50$
TSS:	*Transcription starting site of a gene*
Promoter region:	*±10k bps from TSS*
Window:	*Non-overlapping 200 bps long genome at promoter region*
C_{200}:	*Accumulated read coverage of a window,* $C_{200} = \sum_{i=1}^{200} C_{1,i}$
$\hat{C}_{200}$:	*Approximated accumulated read coverage of a window,* $\sum_{i=1}^{4} C_{50,i}$
s:	*Read intensity of a window,* $\log(\hat{C}_{200})$
b:	*Binary binding indicator: binding event* $b=1$*, non-binding event* $b=0$
a_1:	*Posterior probability for a binding event*
a_0:	*Posterior probability for a non-binding event*
n:	*Index of gene*
w:	*Index of window*
$s_{n,w}$:	*Sample read intensity of* w*-th window at promoter region of* n*-th gene*
$s_{n,w,input}$:	*Input read intensity of* w*-th window at promoter region of* n*-th gene*
μ_{TFBS}:	*Mean of* $s_{n,w}$ *with* $b_{n,w}=1$
σ^2_{TFBS}:	*Variance of* $s_{n,w}$ *with* $b_{n,w}=0$
σ^2_{input}:	*Variance of* $s_{n,w,input}$
$d_{n,w}$:	*Distance between middle point of the* w*-th window to TSS of* n*-th gene*
Δd:	*Window size, 200 bps*
d_p:	*Promoter region length at one side of TSS*
λ:	*Exponential distribution parameter for binding locations*
π:	*Prior probability for a binding event*
Peak:	*One or multiple consecutive windows with posterior probabilities* a_1 *higher than a threshold, i.e. 0.95*
$c_{n,k,j}$:	*Co-regulation event of* k*-th TF and* j*-th TF at* n*-th target gene*
f_{Promoter}:	*Proportion of detected peaks at promoter region among all peaks from the whole genome*
Differentially expressed target gene:	*For single TF, a gene has at least one binding site at promoter region and its differential expression p-value is < 0.05 when the TF is knocked down; For a pair of TF, a common gene has binding sites from both TFs simultaneously at promoter region and its differential expression p-value is < 0.05 when at least one TF is knocked down.*
Average overlap with other method:	*Average value of proportions of genes also predicted by other methods*

References

1. Rozowsky, J., Euskirchen, G., Auerbach, R.K., Zhang, Z.D., Gibson, T., Bjornson, R., Carriero, N., Snyder, M. and Gerstein, M.B. (2009) PeakSeq enables systematic scoring of ChIP-seq experiments relative to controls. *Nature biotechnology*, **27**, 66-75.
2. Kuan, P.F., Chung, D.J., Pan, G.J., Thomson, J.A., Stewart, R. and Keles, S. (2011) A Statistical Framework for the Analysis of ChIP-Seq Data. *J Am Stat Assoc*, **106**, 891-903.
3. McLean, C.Y., Bristor, D., Hiller, M., Clarke, S.L., Schaar, B.T., Lowe, C.B., Wenger, A.M. and Bejerano, G. (2010) GREAT improves functional interpretation of cis-regulatory regions. *Nature biotechnology*, **28**, 495-501.
4. Cheng, C., Min, R. and Gerstein, M. (2011) TIP: a probabilistic method for identifying transcription factor target genes from ChIP-seq binding profiles. *Bioinformatics*, **27**, 3221-3227.
5. Mokry, M., Hatzis, P., Schuijers, J., Lansu, N., Ruzius, F.P., Clevers, H. and Cuppen, E. (2012) Integrated genome-wide analysis of transcription factor occupancy, RNA polymerase II binding and steady-state RNA levels identify differentially regulated functional gene classes. *Nucleic acids research*, **40**, 148-158.
6. 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**, 576-589.
7. Cheng, C. and Gerstein, M. (2012) Modeling the relative relationship of transcription factor binding and histone modifications to gene expression levels in mouse embryonic stem cells. *Nucleic acids research*, **40**, 553-568.
8. Klambauer, G., Schwarzbauer, K., Mayr, A., Clevert, D.A., Mitterecker, A., Bodenhofer, U. and Hochreiter, S. (2012) cn.MOPS: mixture of Poissons for discovering copy number variations in next-generation sequencing data with a low false discovery rate. *Nucleic acids research*, **40**, e69.
9. Ouyang, Z., Zhou, Q. and Wong, W.H. (2009) ChIP-Seq of transcription factors predicts absolute and differential gene expression in embryonic stem cells. *Proceedings of the National Academy of Sciences of the United States of America*, **106**, 21521-21526.
10. Rye, M.B., Saetrom, P. and Drablos, F. (2011) A manually curated ChIP-seq benchmark demonstrates room for improvement in current peak-finder programs. *Nucleic acids research*, **39**, e25.
11. Gerstein, M.B., Kundaje, A., Hariharan, M., Landt, S.G., Yan, K.K., Cheng, C., Mu, X.J., Khurana, E., Rozowsky, J., Alexander, R. *et al.* (2012) Architecture of the human regulatory network derived from ENCODE data. *Nature*, **489**, 91-100.