Supplementary Information

A novel Bayesian network inference algorithm for integrative analysis of heterogeneous deep sequencing data

Yi Liu[1,2,3*], Nan Qiao[1,2,4*], Shanshan Zhu[1,2], Ming Su[1,2,4], Na Sun[1,2,4], Jerome Boyd-Kirkup[1] and Jing-Dong J. Han[1#]

[1] Chinese Academy of Sciences Key Laboratory of Computational Biology, Chinese Academy of Sciences-Max Planck Partner Institute for Computational Biology, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences, 320 Yue Yang Road, Shanghai, 200031, China

[2] Center of Molecular Systems Biology, Institute of Genetics and Developmental Biology, Chinese Academy of Sciences, Lincui East Road, Beijing, 100101, China

[3] School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China;

[4] Graduate University of Chinese Academy of Sciences, Yuquan Road, Beijing, 100049, China

* These authors contributed equally to this work

[#] To whom correspondence should be addressed jdhan@picb.ac.cn

***Keywords*:** Bayesian network; deep sequencing; data integration; epigenomics; histone modification

Content

Supplementary Methods

I. New technical contributions to BN learning methodology

A summary of kernel-based BN learning

Scoring-based approaches are commonly used for inferring Bayesian network structures from data and in particular, there are two opposing forces in a scoring function: one measures the fitness of BN structure to the data and the other one penalizes the model complexity [1]. A typical example of this idea is the commonly used Bayesian information criterion (BIC) scoring function. Suppose the BN structure is represented by a directed acyclic graph G, which has k nodes. BIC is composed of two terms. The first term computes the mutual information (MI) between each node and its parents, and the second one penalizes the number of parameters in the model scaled by the logarithm of the number of training data:

$$S(G) = n\sum_{i=1}^{k} MI(\{\pi_i\},\{i\}) - \frac{1}{2}\log n(\#\, params),$$

where n is the number of training data and π_i denotes the parents of node i.

However, the BIC scoring function cannot be directly applied to model high dimensional probability distributions of sequence tag profiles in its naïve form. This problem can be addressed using the kernel-based BN learning scheme [2], where kernel generalized variance (KGV) [3] is employed as a surrogate measure for *MI*. Here, we generalize such a kernel-based BIC scoring approach to model the tag profiles in deep sequencing data. First, we propose the L1 reciprocal partial sums (L1-RPS) kernel based on the notion of transportation costs for handling tag counts distributions (represented as vectors), thereby enabling this algorithm to perform BN structure learning directly from sequence tag profiles (See sections below). It is worth noting that the capacity of integrating discrete (genotype) /continuous (e.g. gene expression) training data is naturally inherited in SeqSpider because the kernel for these two types of data has already been defined previously [2]. Second, we

developed a ‘wise normalization’ approach, which automatically determines the ‘width’ parameter in the kernels without fine-tuning (See sections below). Third, we intentionally add a free parameter to control the complexity term in the BIC criterion, which allows customized control of the number of edges in the resulting BN structure (thereby obtaining a desired balance between sensitivity and specificity in the final learning results) (See sections below). Finally and importantly, we developed a post BN-learning graph search strategy, which enables us to recover some feedback edges that were missed in the BN structure (See sections below). A summary of the novel functions enabled by the SeqSpider algorithm is listed in Table S1.

Design principles of the novel kernel for sequence tag distributions

The trivial kernel for discrete data and the Gaussian kernel for continuous data have been previously defined [2]. Motivated by the idea of using kernels to summarize data, we developed a kernel to assess the similarity between two sequence tag distributions in order to enable the BN learning algorithm to learn directly from tag count profiles. Since sequence tag distributions are often represented by vectors along the chromosome, the spatial proximity between different bins in the vectors should be considered in the kernel. This is because the short sequence tags are scattered across the genome. If a traditional bin-to-bin similarity measure is used, such as the Pearson Correlation Coefficient (PCC), tags that fall in nearby bins cannot be well matched between two vectored signal intensity profiles. This problem will also be more serious if the sequencing depth is low. On the contrary, if the tag distributions are represented by vectors with very few bins, there will be not much difference if simply learning from the total tag counts in this region. This tradeoff between allowing neighboring tags to match and accurately representing tag distributions is hard to compromise on when using a fixed binning resolution, since tag density could vary significantly over different chromosome regions and across different datasets. As a result, we resort to developing a cross-bin kernel to quantify the proximity between two tag distributions, which is (quasi) invariant to the dimensionality of the vectored profiles.

We propose a Gaussian kernel based on L1 distance of the reciprocal partial sums

of two vectors (L1 reciprocal partial sums (L1-RPS) kernel) (See sections below). The rationale of this construction is two-fold. First, for two sequence profiles with equal total tag counts, the L1 distance of their cumulative distributions is identical to the Earth Mover's distance of the two vectors [4]. As a result, the discrepancy between the 'shapes' of the two intensity profiles is well characterized. Second, for two uniform tag count distributions, it can be shown that the L1-RPS kernel is equivalent to the Gaussian kernel for continuous data (See sections below). Therefore, in the proposed kernel design, both the shape and quantity of the tag count distributions are well characterized. Note that the idea of 'minimal' kernel design [5] in the computer vision literature shares some resemblance with this idea, however, both the mathematics and the purpose of the research are vastly different.

Finally, it is interesting to point out that since partial sums are not sensitive to the binning of vectors, our L1-RPS kernel is robust against the resolution (dimensionality) of tag distributions. As a result, it is especially suited for modeling NGS data sets with varying sequencing depth and is not prone to being affected by the difference in tag density between different chromosome regions.

The L1 reciprocal partial sums kernel for vectored tag profiles

Similar to the definition of Gaussian kernel for continuous variables [2], the kernel for vectored data can be defined as

$$k(x, x') = e^{-d(x,x')^2/2\sigma^2}$$

where $d(x, x')$ is a distance measure for vectors x and x'. The simplest way to complete this definition is to use the Euclidean distance:

$$d(x, x') = \|x - x'\|_2$$

However, as we have shown in the section above, a bin-to-bin distance measure is not suitable for ChIP-Seq signal. Therefore, we define the L1 reciprocal partial sums distance and then use it in the definition of $k(x, x')$.

First, the forward partial sums of the vector $x = (x_1, x_2 \ldots x_n)$ can be represented by

the vector $y = (y_1, y_2 ... y_n)$, where

$$y_i = \sum_{k=1}^{i} x_k \ .$$

Similarly, the reverse partial sums of the vector $x = (x_1, x_2 ... x_n)$ can be represented by the vector $z = (z_1, z_2 ... z_n)$, where

$$z_i = \sum_{k=1}^{i} x_{n-k+1}$$

Then, the L1 reciprocal partial sums (L1-RPS) distance between two vectors x, x' can be defined as

$$d_{L1-RPS}(x, x') = \frac{1}{n+1}(\sum_{i=1}^{n} \left| y_i - y_i^{'} \right| + \sum_{i=1}^{n} \left| z_i - z_i^{'} \right|),$$

where (y, z); (y', z') are the (forward partial sums, reverse partial sums) of x; x', n is the length of the vectors and $|\cdot|$ denotes taking the absolute value of the inside quantity.

Based on the definition of the L1-RPS distance, we can simply insert it into the definition of $k(x, x')$ to complete the definition of L1 reciprocal partial sums (L1-RPS) kernel. It can be easily shown that, for two uniform tag distributions x, x', the L1-RPS kernel is equivalent to the Gaussian kernel [2], where the total counts of x, x' are used instead of vectors.

Finally, it is worth noting that we employed incomplete Cholesky factorization to approximate the Gram matrix in the implementation, as in [3], which not only avoid the necessity of calculating the full Gram matrix, but also effectively filter out any possible negative eigen-components in the matrix. In other words, it is guaranteed that the Gram matrix approximated by the factorization is always positive semi-definite, which does not require the exact positive-definiteness of the kernel itself.

"Wise normalization" automatically determines the widths of kernels

For discrete variables, there is no width parameter in the definition of the trivial kernel [2]. For continuous variable $x_i, i = 1, 2...d_c$, where d_c is the number of real-valued variables in the BN, we first compute the squared average (σ_{x_i}) of the differences for any two instantiations of this variable (x_i^s, x_i^t) in the training data:

$$\sigma_{x_i}^2 = E_{s \neq t}(x_i^s - x_i^t)^2$$

Then, we normalize training data in this dimension by this value so that the squared average of the differences of this variable is a constant:

$$x_i^j \leftarrow 2x_i^j / \sigma_{x_i}, j = 1, 2...m .$$

Here, m is the number of training data. In other words, the width of the Gaussian kernel is automatically resized in proportional to the standard deviation of this variable. The use of the above approach rather than directly computing the standard deviation is mainly for generalizing the normalization approach to other distance measures. Indeed, for the case of vectored variable, the pair-wise L1 reciprocal partial sums (L1-RPS) distances (See the section above for details) were used to quantify the discrepancy between different instantiations of this variable. Accordingly, σ_{v_i} for vectored variable v_i can be defined as:

$$\sigma_{v_i}^2 = E_{s \neq t}[d_{L1-RPS}^2(v_i^s, v_i^t)]$$

Then, the corresponding columns of training data for this variable are normalized by this value so that the squared average L1-RPS distances between any two samples of this vector is a constant:

$$v_i^j \leftarrow \frac{2}{\sigma_{v_i}} v_i^j, j = 1, 2...m$$

Flexible control for the complexity term in the BIC scoring criterion

To enable customized control for the strength of penalty in the BIC scoring criterion,

we modified the second term in the scoring function to the one below (See Eqn. 4 in reference [2] for more details):

$$\lambda \frac{d_{\pi_i} d_i}{2} \log N$$

Here, $\lambda \in (0, +\infty)$ is an adjustable parameter that controls the weight of this term. When $\lambda = 1$, the BIC scoring function is exactly recovered, and d_i, d_{π_i} are the effective dimensionalities of node i and its parents π_i, respectively [2].

Recovering feedback edges missed in the Bayesian network structure

To disambiguate causes and effects, the Bayesian network structure is defined to be a directed acyclic graph (DAG), which strictly prohibits directed loops [1]. However, this is not a reasonable assumption for reverse engineering biological networks, where feedback loops often play essential roles to maintain network stability or for signal amplification. It is therefore helpful to identify potential feedback loops missed in the BN structure.

However, it is not feasible in general for any computational algorithm to predict self feedback loops (e.g., A<->A) and reciprocal feedbacks (e.g., A<->B) purely based on observational data. To identify such relationships, employing interventions (such as using RNAi to change the status of some factors in the system) or measuring gene expression repetitively at a series of consecutive time points is often necessary. (As such, the failure of SeqSpider to identify the H3K4me3<->Oct4 feedback loop is inherently limited by the type of training data)

Despite this general limitation, We have designed and implemented a relaxed, post BN-learning graph search strategy that is capable to identify potential feedback loops with >=3 nodes (such as A->B->C->A) that have been missed due to the acyclicity constraint imposed during BN learning. The basic idea of this approach is as follows. First, we start with the BN structure inferred by the standard graph search algorithm. Then, we continue adding edges to this BN structure in a greedy way with a relaxed graph acyclicity checking module that only prohibits reciprocal and self

feedback loops (such as A<->A or A<->B) in a greedy way to maximize the BN scoring function. In other words, directed loops that involving with >= 3 nodes are allowed. The prohibition for self and reciprocal feedback loops is not an arbitrary or enforced restriction. It is simply because in this way the BN scoring function is still well defined and such very short feedback loops are inherently non-identifiable from observational data alone. Also note that we only allow 'add edge' graph operator to be applied in this feedback-edge hunting stage, since we are not going to modify the BN structure learned in the first stage. Otherwise, the learning results are much harder to explain. Finally, as natural, this procedure stops when the algorithm cannot add any additional edge that will lead to a further increase in the scoring function.

Mathematically, the network we finally learned after the second feedback edge hunting stage is a special, restricted type of dependency networks [6], which allow directed cyclic loops but prohibit reciprocal loops (A<->B). This prohibition makes it fundamentally different with the general dependency networks introduced in that paper. First, we note that the number of feedback edges inferred in the second stage of our algorithm is usually much smaller than the number of edges of the BN learned in the first stage and the strength of feedback edges is also weaker (See Note 19 and Table S9). As a result, the causal structure represented in the BN is basically retained. In contrast, most edges in the general dependency networks are bi-directional (In fact, all edges in a minimal consistency dependency network are bi-directional, as shown by Theorem 4 in [6]), which makes it totally an acausal representation.

The profile-based clustering scheme for preprocessing BN training data

We propose the profile-based clustering approach to reduce technical noise in the training data before BN learning. For the SeqSpider algorithm, all variables (nodes) in the BN, whether continuous valued (e.g., gene expression) or vectored profiles (transcription factors and epigenetic modifications, etc.) are concatenated into a long vector and then executed by a clustering algorithm. For other (real-valued / discrete data-based) BN learning schemes that do not accept vectored nodes, we always concatenate continuous variables before clustering. After that, cluster centers are used

as a new set of representative, less noisy training data for BN learning. The default number of clusters in profile-based clustering is 1000. It is worth noting that in the data re-sampling procedure for deriving the consensus network (See sections below), we always generate 10 subsets of the original gene-wise training data first and then apply profile-based clustering to each subset of training data. Moreover, in the discrete data learning scheme, after profile-based clustering, an additional step of discretizing each node for the 1000 new training cases (cluster centers) into 3 classes is required before BN learning

The above profile-based clustering scheme is fundamentally different from previous data discretization procedures which are required in discrete data-based BN learning algorithms [7, 8], where each node in the BN is only discretized into 2 or 3 classes and all nodes are never concatenated before clustering.

The Super k-means algorithm

In principle, any k-means like algorithm can be used to perform the profile-based clustering procedure above. However, the output clusters of the classic Lloyd's k-means algorithm is often quite unstable, which heavily relies on the initial cluster assignment. To circumvent this problem, we implement a so called "Super k-means" algorithm, which combines two powerful improvements over the classic k-means algorithm: The k-means++ algorithm for initialization [9] and the Hartigan's approach to optimization [10]. The former approach leads to better initial cluster assignment and the latter approach partially mitigates the local optima problem in the Lloyd's iterative optimization procedure. Due to these optimized implementations, the stability and quality of the clustering results are significantly improved (See Note 10 and Figure S6 for more details).

II. General NGS data analysis and network learning methods

Data sets

A set of heterogeneous deep sequencing data (histone modification and TF ChIP-Seq,

DNA methylation bisulfite sequencing and RNA-Seq data) from several laboratories [11, 12] were used to infer the regulatory network of self-renewing hESCs. Specifically, the genome wide DNA methylation BS-Seq data, the 7 histone modification (H3K27ac, H3K27me3, H3K36me3, H3K4me1, H3K4me3, H3K9ac, H3K9me3) ChIP-Seq data and RNA-Seq data were obtained from GSE16256 [11]. The 5 transcriptional factor ChIP-Seq data (OCT4, KLF, MYC, TAFII, P300) were derived from GSE17917, another 2 transcription factor ChIP-Seq data (SOX2, NANOG) were obtained from GSE18292. The 2 polycomb complex ChIP-Seq data (EZH2, RING1B) were downloaded from GSE13084 [12]. See Table S2 for a brief summary of the NGS data used in this study.

Data curation and network learning strategy for the hESC regulator network

In the vectored representation, sequence tags in the chromosome region are summed into 10 evenly spaced bins (10 uniform consecutive intervals) in the TSS [-2kb, +2kb] (or TTS -/+2kb for H3K36me3), which keeps the "shape" of ChIP-Seq/BS-Seq signals. Specifically, the centers of two consecutive bins are spaced by 0.4kb and the width of each bin is set to 0.8kb, so that the estimation of the tag density (denoted by x and over a 20bp unit) in each bin is robust. (The size of the last bin is halved due to the boundary effect, however density estimation is relatively insensitive to bin-size.) For continuous representation ChIP-Seq / BS-Seq data, the tag density x is simply averaged over the entire TSS -/+2kb region. The RNA-Seq data is processed by rSeq [13] to derive the gene expression level (represented by RPKM (Reads Per Kilobase per Million mapped reads) [14]). For the two data types above, we applied the logarithmic transformation ($y = \log_2(x + 10^{-6})$) followed by z-score normalization against the mean and standard deviation over all gene-wise data samples (before generating the 10 subsets of training data, see next section) to standardize the density values in each bin. In the discrete learning scheme, the normalized continuous value of each variable (node) is further discretized into three classes using the k-means

algorithm in the "Clusters 3.0" software (with 100 repeats) before BN structure learning, as described in Yu et al [8].

For discrete BNs, the Bayesian Dirichlet equivalence uniform (BDeu) metric [15] is used as the scoring function. To achieve the best performance, the equivalent sample size (ESS) parameter (which specifies the strength of the uniform Dirichlet prior), is learned using a recursive approach [16]. For continuous data, the Gaussian kernel is employed for each real-valued variable to learn the BN structure in the original kernel-based algorithm [2]. Finally, the BN structures for the heterogeneous data are learned using a combination of the L1-RPS kernel for vectored data and the naïve Gaussian kernel for real-valued variables (Supplementary Methods).

Finally, the single tunable parameter λ in SeqSpider that controls the weight of the penalty term in BIC scoring function (Supplementary Methods) is always set to 3.0 / 4.0 for learning the hESC regulator networks / motif-motif interaction networks, respectively. Good stability of this parameter setting is demonstrated in Note 3.

Data re-sampling scheme for deriving the consensus network

A data re-sampling scheme was used to derive the consensus network and evaluate the stability of different algorithms. Specifically, in the first step, 10 subsets of training data were sampled uniformly without replacements from the original dataset, where every case of training data has an independent, 90% probability to be included in each subset. Therefore, all subsets are different and each contains roughly 90% cases of the original dataset (see [8] for a similar strategy). Then, a Bayesian network is learned on each subset of data. Afterwards, the structure of each BN (represented by a directed acyclic graph (DAG)) is converted to a partially directed acyclic graph (PDAG) to distinguish compelled (directed) and non-compelled (undirected) edges [17]. Finally, the degree of consensus between the 10 PDAGs is quantified by the network stability curve (i.e., the so called Receiver Operator Characteristic (ROC) curve described in [8]), and a consensus network is inferred from the 10 PDAGs (see [8] for the detailed methods). Here, we distinguish two types of network stability curves based on different definitions of edge 'consensus': In the strict sense (directed-edge network

stability curve), overlapped edges connecting two nodes must be the same type (compelled/non-compelled) and have the same directionality if compelled. In undirected-edge network stability curve, only the match between undirected skeletons (see definition below) of PDAGs is considered. We always use the strict definition of edge consensus to derive the consensus network (which consists of edges that appear $\geq c$ times in the 10 BNs, default $c = 7$).

Note that the conversion of DAGs to PDAGs is important for making correct causal interpretations. This is because each BN has a number of equivalent structures (which could not be distinguished based on purely observational training data), which collectively define an equivalence class. All DAGs in this class have the same set of skeleton (undirected connections of the graph regardless the arrows of edges), but the directions of compelled/non-compelled edges are fixed/variable within this class [18]. As a result, only compelled edges represent the set of relationships whose causality can be resolved unambiguously, and non-compelled edges should not be associated with any causal interpretation. A proof of the score equivalence property for the kernel-based scoring function used in this work is given in Note 7, which justifies the validity of the DAG to PDAG conversion and the above causal interpretations.

Validating the quality of the consensus network by literature co-citations

Co-citation analysis is a fast way to extract meta-information from the huge volume of existing literatures by providing an estimation of the association between two scientific terms. To quantify the co-citation of an interaction between two nodes in the network, the co-citation rate (Cr) is defined as: $Cr = \ln(N+1)$, where N is the number of co-cited abstracts for the two nodes, and taking logarithm in the formula is a common practice for avoiding the bias of star genes, which dates back to the definition of the inverse document frequency (idf) term in the seminal tf-idf model in text mining. Here, a co-cited paper means that there exists one or more sentence in the paper's abstract where the nodes themselves or their synonyms co-appear.

Then, the co-citation rate (*Cr*) value of a network is defined as the average of the *Cr* values for its edges. To estimate the statistical significance (p-value) of the *Cr* value for a network, we simulate a random network with the same number of edges 1,000,000 times and compute the probability that the *Cr* value of a random network is equal to or larger than the network itself. Note that before computing the *Cr* value for a network, two nodes (expression and mCGLevel) and the edges linked to them are always removed, since gene expression and DNA methylation are both ubiquitous terms in Pubmed abstracts, their co-occurrence with other scientific terms gives large promiscuous *Cr* value.

In the literature co-citation analysis, the Pubmed abstracts downloaded in Apr. 1, 2011 was served as the text database, where papers were considered if they include genes in *C. elegans*, *Drosophila melanogaster*, mouse and human through homologs defined by NCBI homologene.

The validity of the co-citation result is further examined by looking at the output file generated by the analysis, where the Pubmed IDs and abstracts of the co-cited papers were listed and the co-cited terms were highlighted for review (Dataset S3). In fact, we have performed a systematic evaluation of the literature-mining tool using Gene Ontology and KEGG annotations, which confirmed the high accuracy of the literature co-citation analysis in general (Qiao et al. manuscript in preparation).

Finally, we would like to emphasize that the accuracy of the inferred network is not only validated by literature co-citation, but also by a global validation by the motif-motif interaction networks. Both approaches agree on the best accuracy of the hESC network inferred by SeqSpider (Figure 1C, E), which is yet another evidence to confirm the quality of the literature co-citation analysis.

Data analysis pipeline of the SeqSpider package

The utilities for the analysis of deep sequencing data have been integrated into a coherent pipeline (Figure S24). First, in the 'Seq scan' module, tag distributions are computed by scanning the promoter regions in the genome, which was then used as the training data for the proposed BN inference algorithm. Specifically, for ChIP-Seq /

BS-Seq experiments, the sequence tag distribution signal is often represented by a vector. For RNA-Seq data, the gene expression level is often represented by a continuous value. Discretized data (e.g., SNPs, genotypes) can be incorporated as well. Second, in the 'Spider core' module, 10 subsets of the original training data are randomly sampled without replacements, where every case of the original data has 0.9 probability to be included in each subset. Then, after the (optional) profile-based clustering step for noise reduction, 10 BNs are learned using the proposed SeqSpider algorithm on each subset separately and the structure of each BN (represented by a directed acyclic graph (DAG)) is converted to a partially directed acyclic graph (PDAG) to distinguish reversible and irreversible edges. Finally, the 10 PDAGs are overlapped to extract a robust, consensus network structure and visualized in the 'Layout' module.

Data processing methods for inferring the CD4+ T Cells regulatory network

We re-analyzed the ChIP-Seq data of histone methylations and core transcription factors from CD4+ T cells using the SeqSpider algorithm (as in [8]). In particular, the [-1k, +1k] region flanking TSSs was divided into 5 disjoint, equal-sized (400 bp) bins and the number of tags in each bin was used to model the profiled signal at each promoter. Thus, each ChIP-Seq profile is represented by a 5-bin vector. The curated gene expression data for CD4+ T cells (GSE1133) are also included, which is modeled as a continuous variable. To match the ChIP-Seq datasets with the gene expression dataset, all transcript (refseq) ids were transferred into entrez geneids, as in [8]. Before learning Bayesian networks, each bin of the vectored data was transformed according to the formula $y = \ln(x + c)$, where $c = 9$ is a pseudo count used to avoid 0 tag count. Then, the values in each dimension of the vectored profiles and in the continuous variable were z-score normalized against the mean and standard deviation over all training samples (genes). 90% training data (sampled uniformly without replacements) were used for each round of the 10-fold network learning stage.

Edges appearing equal or more than 7 times out of the 10 resulting PDAGs were selected to constitute the consensus network (Figure S12).

III. Methods for learning the motif-motif interaction networks

The general motif discovery and binding sites identification scheme

For peak detection, SICER [19] was run on the raw histone modification ChIP-Seq data with the following parameters: window size 100, gap size 200 and E-value 1e-5; while MACS [20] was applied to the raw ChIP-Seq data for transcription factors with the default setting. Then, enriched motifs within these peaks were identified by DME2 [21-23] using flanking sequences as background controls. Finally, we use the STORM software [24] (in the CREAD package) to pinpoint the binding sites of each motif in each gene's promoter region (TSS+/-2kb) See sections below for more details.

Finding enriched motifs

We employ the DME2 software [21-23] to find 10 or 11- bp long, conserved sequence motifs that are more enriched in the positive set than the negative set. Here, positive data is defined as the DNA sequences of peak regions detected by the SICER/MACS software. Negative examples are defined to be sequences (1) flanking each extended peak, (2) with the same length of the extended peak and (3) non-overlapping with the other peak regions (Figure S18). Here, extended peak means that the length of each peak is extended symmetrically to at least *n*-bp, where *n* is the length of the 95% quantile when all peaks in this deep-seq experiment are sorted according to the sequence length. As DME2 finds many redundant/overlapping motifs, we visually examined the top 10 motifs generated for each ChIP-seq dataset and only select the 2 or 3 most representative ones for learning the motif-motif interaction network. Here, to avoid obtaining redundant/overlapping motifs, single nucleotide with >5 repeats or double nucleotides with >4 repeats are masked before running the DME2 software. (See Figure S15 and Dataset S1 for these selected distinctive sequence motifs).

Motif detection on promoter regions

The distribution of each motif on the promoter region ([-2kb~2kb] flanking TSSs) is detected by the Storm software [24] (in the CREAD package), where flanking regions ([-4kb, -2kb] and [+2kb, +4kb] flanking TSS) non-overlapping with any other gene's promoter region are used as negative controls. P-value cutoff is set to be 1e-5.

Preparing training data for the motif-motif interaction network

To prepare training data for the motif-motif interaction network, we exploit a strategy similar to the one used for constructing the hESC regulator BN: First, divide each [-2kb, 2kb] region flanking TSSs into 10 disjointed, uniformly distributed bins and count the number of motif occurrences within each bin. The result of this step is a 10-dimensional vector that characterizes the occurrence of each motif in the promoter region of each gene. Then, raw tag counts x in each bin are logarithmically transformed according to the formula $y = \ln(x+1)$ and z-score normalized over all training samples (genes). For learning motif-motif networks using continuous data, we use the same data processing approach as described above except we treat the data as 1-dimensional (1-bin) vectors. Finally, the presence or absence of each motif in the -/+2kb region around each TSS was used to generate the 1/0 binary training data for learning discrete BNs.

Learning the motif-motif interaction networks

Given the positions of sequence motifs in a genome, there are basically two ways to infer their relationships. The first is to learn a BN simply based on the co-occurrence pattern of these motifs without *a priori* knowledge of the DAG structure. By applying the proposed vectored data BN learning algorithm based on "L1-RPS kernel" and by deriving the consensus network using the 10-fold data re-sampling scheme, we obtain an unconstrained network for all the sequence motifs (Figure S11).

Although this unconstrained network identifies nearly all potential relationships among these motifs, many of them are false positives. Some of the false positives

might be true positives under other cellular contexts that are unrelated to hESCs, as no context dependent information was included and only the full genome sequences were examined (See Figure S19 for the shortest distance distributions of all motif pairs as independent evidence of potential motif-motif interactions). The interactions within the hESC regulator BN (Figure 1A) can be used as a structural prior to constrain the space of graph search in learning the motif-motif BN, which enables us to uncover the mechanism underneath the *de novo* inferred hESC regulator BN. Specifically, only the motif pairs that are derived from a connected node pair in the hESC regulator BN are allowed to have an edge in the motif-motif BN. Thus, the motif-motif BN can be learned more specifically, more accurately and much faster. After the 10-fold network learning procedure, we obtain the final consensus motif-motif interaction network from this hierarchical learning framework (Figure S22a).

Finally, to make the size of constrained / unconstrained motif networks roughly comparable, we always use the default threshold of edge consensus ($c = 7$) for constrained motif network and a stricter definition of edge consensus ($c = 8$) for unconstrained motif networks (See section "Data resampling scheme for deriving the consensus network" for more details).

Estimating the significance of motif networks by motif-motif proximity

To validate the significance of motif networks by the spatial proximity of motif-motif interactions, we start by introducing a way to compute the distribution of chromosome distance for a pair of motifs. First, the occurrences of each motif in the promoter regions ([-2kb, +2kb] flanking TSSs) are detected. Then, duplicate copies of a motif occurrence that were mapped by different promoters to exactly the same genomic position are eliminated, since otherwise they would be counted multiple times, making subsequent analysis biased. After that, we compute the distribution of the shortest distances for each motif pair. Specifically, for sequence motifs (m_A, m_B), we first compute the distribution of the distances from the nearest m_B position to each m_A position in the genome. Here, only motif pairs within 4kb distance are counted.

Reversely, the distribution of the distances from the nearest m_A position to each occurrence of the m_B motif in the genome is computed, also within the 4kb maximum range. The median distances of the above two distributions are denoted by $D_{B->A}$ and $D_{A->B}$, respectively, and the shortest distance distributions for all motif pairs are shown in Figure S19.

In order to validate the quality of a global motif-motif network, we estimate two distributions. The first one (positive distribution) is the median distance distribution for motif pairs that are connected by an edge in this network, and the second one (null distribution) is the distribution of median distances for all motif pairs in the network (as if the network is fully connected). As mentioned above, the directionality of edges is not considered here and hence two asymmetric median distances are considered for each motif pair. The results for the general motif interaction network and the hESC context-specific motif network are shown in Figure S23.

To estimate the significance of deviation of the positive from the null median distance distribution, one-sided Student's (Welch's) *t*-test was performed.

Estimating the statistical significance of network overlaps

To estimate the statistical significance of the overlap between two motif-motif interaction networks (N_A, N_B), we compute the number of edges in four sets: (1) all possible edges between the union set of nodes of the two networks; (2) the edges in network N_A; (3) the edges in network N_B; (4) the overlapped edges between the two networks. Then, Fisher's exact test can be used to test the statistical significance of the overlap between the two networks. (Note that the directionality of edges is not considered in computing the network overlaps)

To estimate the significance of the overlap between a motif interaction network and a hESC regulator network, we first map all motif nodes to the corresponding nodes (transcription factors/epigenetic modifications) in the regulator network (edges are transformed accordingly). Second, we also remove nodes (and the corresponding

edges) in the hESC regulator network that do not have a counterpart in the motif network (e.g., gene expression level) to make sure that we compare networks among the same set of nodes. Finally, we perform Fisher's exact test between the two networks in order to estimate the significance of their overlap. Again, directionality of edges is not considered during this process.

Supplementary Notes

1. Traditional algorithms for learning Bayesian network structure

Bayesian network structures correspond to a specific factorization of the joint probability distribution of all nodes (variables) to many local probability terms, each only involving the conditional distribution of a node given all of its parents (Local CPDs). In traditional Bayesian network learning algorithms, it is often assumed that each node is represented as a discrete variable. Therefore, the local CPD is often parameterized as a multinomial distribution given each combinatorial instantiation of its parents. By assuming a joint Dirichlet prior for the multinomial distributions and then integrating out the parameters that define these distributions, the marginal posterior probability can be computed for all possible Bayesian network structures (i.e., DAGs), which can be used to search for the best BN structure in terms of its fitness with training data [15]. This scoring-guided search scheme is theoretically sound and it is perhaps the most widely used approach to inferring Bayesian network structures from data, where the particular marginal probability-based scoring scheme above is often called the Bayesian Dirichlet equivalence (BDe) metric [15]. If we assume that all nodes (variables) in a BN are continuous and their joint probability distribution is a multivariate Gaussian, there is a counterpart of the BDe metric, namely the Bayesian Gaussian equivalence (BGe) scoring function [25].

Since the factorization of the joint probability distribution in Bayesian network representation corresponds exactly to a set of conditional independency relationships defined by the graph [1]. A different class of approaches to learning Bayesian network employs conditional independency tests to eliminate the graph structures that are not consistent with training data. One of the most well-known algorithms in this class is perhaps the PC algorithm, which is elaborated in a book [26]. Again, technically it is only tractable to learn discrete and linear-Gaussian Bayesian networks, where the asymptotic Chi-square test corresponds to the former and the Fisher's Z test for the

latter [26].

Currently, when inferring Bayesian networks from biological datasets (e.g., [7, 8, 27, 28]), one needs to convert a continuous or a richer structured signal, such as the complex patterns of histone and DNA modifications at promoters and other genomic regions, to binary or categorical quantities, suffering unnecessary, severe information loss. This is because most existing BN learning algorithms only support modeling the interactions between discrete variables or between real-valued variables that are jointly multivariate Gaussian distributed (which is seldom true for real-world biological datasets), as we have mentioned above. To overcome this important limitation, we developed the SeqSpider algorithm, which enables modeling free interactions between heterogeneous types of variables (discrete/real/profile) simultaneously in one Bayesian network.

2. The unique advantage of SeqSpider against alternative BN learning schemes

To demonstrate the effectiveness of the SeqSpider algorithm to biological applications we compared the performance of its modules against existing methods and algorithms. As the SeqSpider algorithm could work on a hybrid of vectored data (tag distributions) and continuous data (the gene expression level), we use the corresponding kernels for the two types of data and the Super k-means clustered profiles for learning the hESC regulator network (see sections above). We also compared SeqSpider to other algorithms that are based on discrete data/continuous data, rather than a hybrid of vectored data, and also to algorithms that do not use Super k-means clustered profiles. In the data re-sampling scheme for deriving the consensus network, we adopted the network stability curves to measure the robustness of a BN learning algorithm on partial training data (see sections above). We further evaluated the biological relevance of the BN by literature co-citation rate, which represents the frequency of two predicted interactors co-cited in the PubMed abstracts (see sections above).

Compared with the performance of conventional discrete data-based BN learning

(Figure 1B, panel a), using real-valued data (based on the conventional "Gaussian kernel" according to [2]) or using vectored data (based on the "L1-RPS" kernel we developed) only slightly improved the network stability (see Figure 1B, panel b and c for the corresponding curves and the associated AUC (area under curve) values), and was not sufficient for robust inference of BN structures.

When profile-based clustering (by the Super k-means algorithm) was introduced as a preprocessing step prior to BN learning, compared with the discrete or continuous data-based BN learning scheme (Figure 1B, panel d, e), we observed a remarkable improvement in the performance of the "L1-RPS kernel" approach based on tag distributions (represented by vectored data), where both the accuracy and coverage of the corresponding curve approached nearly 100% (Figure 1B, panel f).

We also tested a relaxed approach to computing the network stability curves, where the directionality of the 10 PDAG's edges is no longer distinguished (i.e. only considering the *undirected* skeleton [1] of each BN). The resulting curves of the six learning schemes are also shown in Figure 1B. Clearly, although the curves of all approaches are improved due to this relaxation, the "Super k-means profile-based clustering" plus "L1-RPS kernel" method is still one of the best BN learning schemes in terms of stability (Figure 1B, panel f).

Since profile-based clustering of input data has a big impact on the vectored-data based BN-learning scheme, we compare the performance of this BN-learning method on cluster centers generated by Super k-means (Figure 1A and Figure 1B, panel f) with those by two other representative clustering algorithms, the classic k-means algorithm in Cluster 3.0 [29] and the affinity propagation algorithm [30]. From Figure 1A and Figure S5, the two networks derived from the affinity propagation and the ordinary k-means algorithm (ap.vec, kmeans.vec) are approximately sub-parts of the network inferred from the Super k-means cluster centers (sk.vec) (except one edge H3K4me1-TAFII in ap.vec). The existence of a core sub-network structure convincingly demonstrates the robustness of the profile-based clustering plus the vectored data-based BN learning scheme. However, we can see that the numbers of consensus edges inferred by the two alternative clustering algorithms are much

smaller than the Super k-means algorithm, which suggests the latter has the best noise reduction performance when used for data preprocessing, since edges with weak data support will be overwhelmed by noise in those alternative clustering algorithms.

Indeed, when evaluated quantitatively, the network derived from Super k-means algorithm (sk.vec) significantly excels both the two alternative clustering algorithms (kmeans.vec, ap.vec) on literature co-citation P-values and on network stability curves (Figure 1C, Figure S5, Figure 1B, panel f). Moreover, when examining the overlap between the three hESC BNs (sk.vec, ap.vec, kmeans.vec) with the three general motif BNs (motif.vec/dis/real.uncons, see below), the "sk.vec" BN also has the best overlap with the three motif BNs (Figure 1E). The added improvement of the Super k-means algorithm, over that of k-means, is perhaps because the clusters generated by Super k-means are much tighter than those by k-means, as measured by the sum-of-squared Euclidean distances (SSD) from each data to its cluster center (Note 10 and Figure S6). The bad performance of the affinity propagation algorithm may be attributed to the fact that the cluster centers generated by exemplar-based clustering algorithms are always a subset of the original data (exemplars). As a result, noise in the data is not reduced [30].

In fact, the literature co-citation analysis and motif-regulator network comparison strategy above is not only used for comparing different profile clustering algorithms. We have applied the two quantitative network evaluation approaches to all the eight hESC networks derived from different BN learning schemes (see Dataset S2). First, in terms of literature co-citation P-values, the network generated by SeqSpider (sk.vec) is best supported by existing biological literatures. All the other traditional BN learning schemes based on either discrete or real-valued data did not even generate a regulatory network that has statistically significant literature co-citation rate (Figure 1C).

To further investigate the accuracy of different network inference schemes, we learn three cellular context independent, genomic sequence-based motif interaction networks using discrete, continuous and vectored data-based BN learning algorithms, and then exhaustively compare them with the eight hESC regulatory networks (Figure

1E, Supplementary Methods). Here, profile-based clustering is not used to preprocess the motif data before BN learning as the occurrence pattern of sequence motifs in the genome is exact, which does not require an extra noise-reduction step. Otherwise, not only the motif occurrence signal will be blurred but also the number of training cases will be insufficient to infer a large motif interaction BN.

Among the overlaps between the eight hESC epigenome-based regulator BNs and the three general motif interaction BNs, the hESC regulator BN learned using SeqSpider (sk.vec, Super k-means profile-based clustering plus vectored data-based BN learning algorithm) has the best overlap with the three motif interaction BNs (two of the overlaps are statistically significant). In contrast, except kmeans.vec (the one most similar to sk.vec), all other hESC regulatory BNs inferred by other means could at most match well with one motif BN, with much less significant P-values (Figure 1E).

Reciprocally, among the three motif BNs, the vectored data-based general motif BN also has the best overlap with the hESC regulator BN learned using SeqSpider (sk.vec, see Figure 1E). Again, this fact suggests that compared with discrete or real valued data-based BN learning, vectored data-based learning is also the best for motif BN inference. Therefore, we also inferred a hESC context dependent, constrained motif-motif interaction network using vectored data to best uncover the motif interactions underpinning the hESC regulator network (Figure S22a, Supplementary Methods).

The multiple, independent validations of the epigenome-based hESC network are summarized in Figure 1D. First, the pure genomic sequence-based motif interaction BN (motif.vec.uncons in Figure 1E, see Figure S11 for details) has a statistically significant overlap (Fisher's exact test P=0.0067) with the hESC regulator interaction network (shown in Figure 1A). Second, despite study bias and incompleteness in the literature, the interactors inferred from the hESC epigenome are significantly more frequently co-cited than expected by random (empirical P=0.017, Figure 1D). Third, the constrained, hESC context-dependent motif interaction network (Figure S22a) also has a statistically significant overlap with the unconstrained motif network

(P=6.15e-8). Both the unconstrained and constrained motif interaction networks are validated by the significant spatial closeness of interacting motifs in the promoter regions (one-sided Student's t-test P=2.18e-26 and 1.08e-26, respectively).

To conclude, these quantitative evaluation results solidly demonstrate the unique advantage of combining the Super k-means based profile clustering with the vectored data-based BN learning approach for inferring the hESC regulator network. Since the L1-RPS kernel is at the heart of SeqSpider algorithm for modeling and comparing sequence tag profiles, we also tested two other ways to define the kernel for vectored data [31]. Again, experimental results demonstrate the performance of the L1-RPS kernel excels these alternative methods (Note 13, Figure S7).

Finally, network stability curves also demonstrate the hESC regulator network inferred by SeqSpider is very robust to the choice of the single customizable parameter (the strength of the model complexity penalty in the scoring function) (Figure S8, Note 3) and the internal parameter (the width of kernel) (Note 11, Figure S9). More importantly, we also demonstrate the network model we learned is not over-fitted to the training data (Note 12, Figure S10, Table S4, S5). In summary, SeqSpider outperformed all existing algorithms, when evaluated by network stabilities, literature co-citation rates and other criterions.

3. The hESC regulator network is robustly predicted against the adjustment of regularization strength

One of the major problem for inferring regulatory networks from deep sequencing data is that different nodes in the network are often weakly correlated despite that there is often no genuine molecular interaction between them. This phenomenon could be partly attributed to the background noise in high-throughput sequencing, which is hard to avoid completely. To overcome this problem, customized tuning of the regulation strength is often necessary for reliable network inference. As discussed in Supplementary Methods, we achieve this goal by introducing a parameter λ in the BN scoring function to balance the weight of the complexity term with likelihood.

It is hoped that when this parameter is set appropriately, false positive edges in the BN could be eliminated as much as possible while preserving most true edges.

To demonstrate that λ is properly set and the hESC regulator network is robust inferred, we derive consensus networks using the 10-fold data re-sampling scheme under a set of λ values around the actual instantiation (3.0). Then, the discrepancy between consensus networks derived from different λ is characterized by a network stability curve, which offers a direct assessment of the network stability against changes of the regularization strength (Figure S8). The results show that for a wide range of λ [2.0~4.0], the consensus networks are nearly identical, which strongly supports that true molecular interactions are correctly detected and are well separated from background noise at $\lambda = 3.0$. In other words, good stability of network inference against adjustments of regularization strength is demonstrated. With a similar argument, $\lambda = 4.0$ is used for all the motif network inference experiments.

4. Biological significance of the hESC regulatory network

To evaluate the biological significance of SeqSpider algorithm for *de novo* prediction of regulatory relationships, we looked in more detail at the hESC regulator network learned from the heterogeneous hESC datasets (Figure 1A). In this network, the data from different laboratories and different cell lines are fully intermingled according to their biological relationships. The algorithm inferred that H3K4me3, which directly correlates with gene expression, sits at the center of the graph, connecting six domains in the consensus network: 1) the NANOG domain, which consists of Nanog, Sox2 and TAFII, with the former two factors known to bind each other; 2) the Polycomb domain, which consist of two polycomb genes RING1B and EZH2, the EZH2-catalyzed H3K27me3 modification and its mutually exclusive modification H3K9ac; 3) the enhancer domain (H3K4me1 and H3K27ac, which specifically marks enhancers [32, 33]); 4) three other separate domains: the DNA methylation domain (mCGlevel), the Oct4 domain (Oct4, Klf4) and the Myc domain (Myc). Overall, both the domain structures and 87.5% of the interactions learned from our BN approach have been reported in the literature (except for only two interactions, the one between

H3K27me3 and H3K9ac, and the one between NANOG and TAFII).

In particular, transcription factors Nanog and Oct4 play important role in the early development of embryonic stem cells [34]. Nanog, Sox2 and P300 were found collaboratively regulating gene expression, which was supported by several studies [35, 36] and it is known that Klf4 act as an upstream regulator of Oct4 [37]. Recent studies suggest that both H3K4me1 and H3K27Ac were specifically enriched around enhancers [32, 33]. In line with the notion that DNA methylation was a long term gene repression signature compared with histone modifications, H3K4me3 was found inversely correlated with mCG [38] (See Figure 1A). Myc is a key transcription factor in cellular proliferation, and was found to play an important role in Pol II pause release of a large group of transcriptional active genes in embryonic stem cells [39], which might explain why it is in a domain separated from the core gene expression regulatory factors. Finally, it is known that both H3K4me3 and H3K9Ac are related with transcriptionally active genes, while H3K27 tri-methylation is related with transcriptionally repressed genes [40].

Moreover, it is good to see that although the data for EZH2 and RING1B and that for H3K27me3 were generated from different laboratories and from different cell lines, the biological relationships between EZH2, RING1B and H3K27me3 were still clearly depicted in the network (Figure 1A).

The network is also successful in integrating different types of training data: the expression for each gene is represented as a real value, while the other data sets (DNA methylation/histone modification/TF/PcG protein binding) are represented as 10-bin vectors. Despite the heterogeneity of the input training data, SeqSpider still correctly identifies H3K4me3 and H3K36me3 as factors that were most closely associated with gene expression [40, 41]. It is interesting to see that the network predicts nearly all the other modifications, and the investigated TF/PcG binding, link to gene expression through H3K4me3, which is certainly true for PcG proteins [12] (Figure 1A). Besides this fact, researchers have recently demonstrated experimentally a positive feedback interaction between H3K4me3 and OCT4 [42]. The lack of irreversible edges in this network is not because SeqSpider cannot learn causal relationships. On the contrary,

SeqSpider can accurately recapture most of the causal relationships inferred by a conventional BN learning algorithm on a CD4+ T cell ChIP-Seq dataset, and correctly identified other relationships missed by the conventional algorithm (Note 14, Figure S12), which is a strong evidence that SeqSpider can disambiguate causalities at least as well as a standard BN learning algorithm. In addition, we have established, theoretically, the correctness of SeqSpider algorithm for exploring causal relationships within the BN formalism (Note 7). By incorporating more recent ChIP-Seq data on hESC, some directed edges could be added to the network, but the key structure of hESC network is retained with only a few refinements/changes, further demonstrating the robustness of SeqSpider algorithm against fluctuations in the data (Note 15, Figure S13). Finally, we would like to remark that the global "star-shaped" topology centered at H3K4me3 in the hESC regulator network has further implication in the directionality or causalities for the interactions. This point will be discussed with more detail in Section "Globally consistent causal interpretations of the hESC regulator network".

By separately inferring BN for different groups of transcription factor and histone modification profiles, we found that the consensus hESC network in Figure 1A is the most comprehensive representation of the relationships inferred from different groups of profiles (Note 16, Figure S14). Moreover, even using a small subset (~40%) of randomly selected training data, we can still predict the structure of the consensus network very well (Note 20, Table S10).

Of course, there are still a few known regulatory relationships that were not predicted in the hESC regulator network. However, careful analysis suggests these interactions were either feedback edges that are prohibited by the acyclic constraints of BNs, or were not well supported by the deep sequencing datasets used for network inference (Note 18, Table S6). To retrieve those feedback edges missed in the network, we designed a post BN-learning graph search strategy (Supplementary Methods), which recovered some of the known interactions missed due to the acyclic constraints in the BNs for deriving the consensus hESC regulator network (Note 19).

In summary, we have demonstrated that the hESC regulator network produced by

SeqSpider suggested an important biological phenomenon, which was also supported by other existing data. This implies the potential of employing SeqSpider for *de novo* identification of biological interactions/regulations in other cellular contexts.

5. Globally consistent causal interpretations of the hESC regulator network

Inferring causal relationships from biological datasets is a longstanding challenge in computational biology. To this end, Bayesian network has been extensively used in systems biology research. However, the conditional independencies embodied in the pure observational training data only enable BNs to resolve the causal ambiguity for some of the edges (a.k.a. compelled edges) in the network, while left the directionality of the other edges (non-compelled edges) unspecified due to the structure equivalence between different BNs. Moreover, BNs could not handle feedback loops due to the acyclic assumption of the graph structure by definition. Overall, these limitations are general to the type (observational) of training data and to the BN formalism, which is not specific to SeqSpider.

In our case, one striking feature of the hESC regulator network (Figure 1A) is that all edges in this network are non-compelled, which means that locally the directionality of each edge could not be inferred from data [18]. However, this is not to say that the directions of the 16 edges in this network can be oriented arbitrarily, as in a trivial undirected network. In fact, there is strong causal regularities in the global network level: The consensus hESC network is rather stable that it appears exactly as is in five out of the ten PDAGs in the data re-sampling procedure for inferring the consensus network, it allows us to derive globally consistent causal interpretations.

Specifically, the network structure has a perfect "star" topology with H3K4me3 at the center and all other nodes are located at the 7 branches connected to H3K4me3. If two or more out of the 7 edges that linked to H3K4me3 are oriented inwards, then one or more v-structures (immoralities) will inevitably be created, which makes the corresponding edges compelled (fixing the directionalities) [18], contradicting with

the global non-compelled topology of the network. Therefore, either all the seven edges are oriented out of H3K4me3, or only one of them is allowed to orient inwards (Figure 1F).

Given the central causal topology specified, the directionality of almost all edges in the network could be determined using the Meek's rules [17]. In fact, when an edge connected to H3K4me3 is oriented outwards, this orientation will be propagated to all edges in this branch. Therefore, we obtain eight global causal configurations of the whole network which corresponds to the eight central topologies above, as shown in Figure S17.

These globally consistent causal interpretations of the hESC regulator network prompts an intriguing promoter-enhancer interaction model, where H3K4me3 serves as the dynamic, input/output information relaying hub at the center. Once it receives an stimulation from one promoter/enhancer domain, it will propagates information to other promoter/enhancer domains. Thus, the initial signal for transcription at any domain could be amplified for a gene. In support of this hypothesis, one of the interactions around H3K4me3, between Oct4 and H3K4me3, has recently been demonstrated experimentally to be bidirectional, and feedback nature is essential for self-renewal and pluripotency of hESC [42].

6. Molecular mechanisms and biological significance of the hESC context-specific motif network

The utility of SeqSpider is further demonstrated when we integrate additional data to interpret the molecular mechanisms in the hESC regulator network. Specifically, we infer a motif-motif interaction network which is consistent with the hESC regulator network in order to explain the sequence motifs that potentially mediate the regulator interactions in the hESC context. This was achieved by using a constrained BN learning algorithm, which takes the regulator network as a template (Methods and Figure S22). In this hESC context-specific motif network, over 50.9% of edges overlapped with the unconstrained motif-motif network (with Fisher's exact test P=

6.15e-08). Both the unconstrained (general) and constrained motif interaction networks can be further validated by the spatial closeness of motif locations in the promoter regions, which were not used in BN inference (One side t-test P=2.18e-26 and 1.08e-26, respectively, See Figure S23 and Supplementary Methods).

By learning this hESC context-specific motif network, additional molecular interacting mechanisms can be further derived for the hESC regulator network. For example, we can infer that the Oct4-H3K4me3 relationship occurs through the interactions of two pairs of binding motifs: [CCGCCGCCGCC] (H3K4me3_1_len11) - [CCTCCCCGCCC] (OCT4_1_len11), [CCCCGCCCCC] (H3K4me3_5_len10) - [CCTCCCCGCCC] (OCT4_1_len11). Another example involves the interactions between TAFII, NANOG, SOX2 and P300. As shown by the motif interaction network, the two motifs of NANOG have different functions for mediating the interaction between NANOG and other hESC regulators: First, TAFII regulates NANOG through [GGCCCCGCCCC] (TAFII_4_len11) -> [TTCAAATGCAA] (NANOG_6_len11). Then, NANOG regulates SOX2 through [TTCAAATGCAA] (NANOG_6_len11) -> [AAATGCAAAT] (SOX2_5_len10). Afterwards, SOX2 in turn regulates NANOG through motifs [AAATGCAAAT] (SOX2_5_len10) -> [AAATTTGCAT] (NANOG_1_len10). Finally, NANOG regulates P300 through [AAATTTGCAT] (NANOG_1_len10) -> [GAAATGCAAAT] (P300_1_len11) (See Figure S22a and Figure S15, referring to both the hESC context-specific motif network and the motifs).

Here, the Nanog's interaction with TAF II might suggest its direct association with the core transcription machinery. The motif interactions between Nanog, Sox2 and P300 clearly explained how the three transcription factors collaborate at the sequence level with high confidence. Although motif interactions may imply cooperative targeting, rather than TF-target interactions, for pinpointing which ChIP-Seq derived BN interactions correspond to cooperative targeting, the fact that many of the regulators are self-regulating and inter-regulating to form intricate feedback control circuits [34] may also implicate the transcriptional regulations represented by the inferred motif-interactions. For example, P300 was found directly

linked to Nanog expression in mouse embryonic stem cell [35], Nanog RNAi in human embryonic stem cell resulted in an 82% decrease in Sox2 expression [36] and knockdown of Sox2 could in turn decrease Nanog expression.

It is interesting to point out that although motif pairs identified in the network have much smaller chromosome distance in general, our BN learning approach also predicted some rather long distance motif-motif interactions (EZH2_5_len11 vs. RING1B_5_len11 and TAFII_4_len11 vs. NANOG_6_len11). For the former one, it is well-known that EZH2_5 and RING1B are both members of the Polycomb group complex. It would be interesting to see if these interactions can indeed occur within rather long distance, although here we limited our distance measurements for any motif pair within 4 Kb. That the proposed BN learning approach has the potential to predict long range motif-motif interaction is an apparent advantage over a simple approach based solely on motif-distance analysis, which would inevitably miss such long range motif interactions.

7. A Proof of the score equivalence property for the kernel-based scoring function

As we have mentioned before, all BNs in an equivalence class represent the same set of conditional independency relationships. It is therefore important for an ideal BN learning algorithm to assign the same score to the BNs, so that no spurious causalities are artificially added by the algorithm. This desirable property of BN scoring function is called score equivalence, which we shall prove it for the SeqSpider algorithm.

The kernel-based scoring function for BN structures is defined by Equation (4) in reference [2] (We only added a parameter λ to control the weight of the second term). Here, we prove this scoring function is score equivalent, justifying the usage of the causal inference approach [17] for the interpretation of BN structures with/without structural constraints.

Based on Theorem 2 in reference [18], it suffice for us to prove the kernel-based scoring function is invariant for two BNs which simply differ by the reversal of a

covered edge [18].

Denote the graph of the two BNs by G and G', the covered edge in G by $i \to j$, which is reversed to $j \to i$ in G', and the parent sets for nodes i, j are indicated by π_i, π_j respectively. Based on the definition of covered edge, $\pi_j = \pi_i \cup \{i\}$ in G. Accordingly, the total score of the two nodes' families in G is:

$$J(i,\pi_i) + J(j,\pi_j)$$
$$= \frac{N}{2}\left[\log \frac{\left|R_{\pi_i \cup \{i\}, \pi_i \cup \{i\}}\right|}{\left|R_{\pi_i,\pi_i}\right|\left|R_{i,i}\right|} + \log \frac{\left|R_{\pi_i \cup \{i,j\}, \pi_i \cup \{i,j\}}\right|}{\left|R_{\pi_i \cup \{i\}, \pi_i \cup \{i\}}\right|\left|R_{j,j}\right|} \right] + \lambda \log N \left[\frac{d_{\pi_i} d_i}{2} + \frac{(d_{\pi_i} + d_i) d_j}{2} \right]$$

After the reversal of this edge, the family of node i becomes $\pi_i^{'} = \pi_i \cup \{j\}$, the family of node j becomes $\pi_j^{'} = \pi_i$. Now the total score of the two nodes' families in G' is:

$$J'(i,\pi_i^{'}) + J'(j,\pi_j^{'})$$
$$= \frac{N}{2}\left[\log \frac{\left|R_{\pi_i \cup \{j,i\}, \pi_i \cup \{j,i\}}\right|}{\left|R_{\pi_i \cup \{j\}, \pi_i \cup \{j\}}\right|\left|R_{i,i}\right|} + \log \frac{\left|R_{\pi_i \cup \{j\}, \pi_i \cup \{j\}}\right|}{\left|R_{\pi_i,\pi_i}\right|\left|R_{j,j}\right|} \right] + \lambda \log N \left[\frac{(d_{\pi_i} + d_j) d_i}{2} + \frac{d_{\pi_i} d_j}{2} \right]$$

By re-arranging and canceling redundant terms, it is easy to show that $J(i,\pi_i) + J(j,\pi_j)$ is equal to $J'(i,\pi_i^{'}) + J'(j,\pi_j^{'})$. Since the score of the BN structure is simply the sum of the scores of each node's family, we have proved that the kernel-based scoring function is invariant to covered edge reversal (and therefore the score equivalence property of this function regardless the specific kernel definition).

8. Further discussion about the hESC regulator network

We have demonstrated the biological significance of the hESC regulator network at length in Note 4. In this section, we present some finer grained discussions about this network, which might be helpful for potential users to gain a deeper understanding of how the limitations of NGS data and the BN formalism affect BN learning, how we could mitigate some of these problems. We also discuss the potential applications of SeqSpider in other biological studies.

Although H3K9me3 is known to precede DNA methylation in many cases, we failed to learn any relationship for H3K9me3, most likely due to its insufficient sequencing depth (Figure S16, Table S7, S8). The lack of the known interaction between P300 and SOX2 could be explained by the fact that we already have two edges in the hESC regulator network (SOX2-NANOG, NANOG-P300), and hence the addition of P300-SOX2 may create a loop in the network structure, which is prohibited by the acyclicity constraint of BNs (Figure 1A, Table S6). In fact, using the proposed post-BN learning feedback edge hunting algorithm, not only P300-SOX2, but also three other interactions between the hESC / enhancer marks to the transcription initialization domain are recovered, which further refined the hESC network (Note 19, Figure S20, Table S9).

Although the regulatory network we present here was inferred for hESC, such network topology and gene expression regulatory mechanism is likely to function under other cell contexts, as all epigenetic modifications are ubiquitously present. As the application of SeqSpider is not restricted to hESCs, and should be applicable to many other experimental systems, different cellular context-dependent BNs can be learned when the data are available. Currently, we have inferred a similar regulator network from NGS data of mouse embryonic stem cells (mESC) and demonstrate a similar network topology and the role of H3K4me3 as the information relaying hub in the mESC network (Note 17). Moreover, we have also predicted a regulatory network for CD4+ T cells (Note 14). As to the potential future application of SeqSpider, we conjecture that when precise genome-wide maps of enhancer sites become available, SeqSpider can be used to explore the long distance interactions between enhancers and TSSs (Note 21).

Finally, it is interesting to point out that SeqSpider can correctly infer cell-type specific features of regulatory networks from NGS datasets. For example, we have shown that there is no irreversible edge in the regulatory networks inferred for self-renewing hESC and mESC cells, while many irreversible edges exist in the regulatory network predicted for CD4+ T cells. Besides, H3K4me3's role as the hub in the hESC/mESC network is perhaps not true for the CD4+ T cell network. This

observation implies that the hub role of H3K4me3 is dynamic in ESCs: it can either receive or propagate information between key ESC regulators. The specific feature in ESCs, in part, has recently been experimentally demonstrated by a Cell paper, which confirmed the existence of a positive feedback loop between OCT4 and H3K4me3 in hESCs, which is essential for maintaining the high level expression of hESC-specific key TFs [42] and for pluripotency maintenance. As a result, that edges connecting H3K4me3 in the hESC/mESC network are reversible is consistent with biological knowledge. This difference is therefore likely to reflect the diverse regulatory program in the two cell types. The feasibility of using SeqSpider to infer cell-type specific features of regulatory networks is clearly demonstrated through this example.

9. Data transformation and bin size for learning the hESC regulator network

The intuition behind applying the logarithm and z-score transformation to raw tag density distributions in the +/-2kb promoter region around TSSs prior to BN learning is illustrated in Figure S3. Due to biological and experimental variations, for different epigenetic modifications / transcription factors, the distributions of raw tag density in the [-2kb, 2kb] promoter region across different genes are highly skewed and differ a lot between each other (Left column). Therefore, it is hard to infer an unbiased regulatory network directly from such data. To overcome this problem, we first apply logarithm transform to the raw tag density values and it is easy to see that the shapes of the distributions become more regular and less skewed (Middle Column). However, variations in the widths and centroids of these distributions are still quite large, which prompts us to apply the z-score normalization afterwards. It is easy to see that after the latter normalization, discrepancies in both the widths and centroids of these distributions are greatly reduced (Right Column).

It is also important to choose the most appropriate bin size for representing tag profiles. To this end, we perform the BN learning procedure using the same method for the curation and normalization of profiled data, except the tag count distributions

in the [-2kb, +2kb] region surrounding TSSs are represented by different number of bins. Specifically, in addition to using 10-bin vectors to represent tag profiles for learning the hESC regulator network in the manuscript (Figure 1A), we have also tried using 6, 8, 12 or 14-bin vectors for the same task. All parameters and methods for network inference and for deriving the consensus network are left unchanged. The consensus networks derived from the four new experiments (Dataset S4) are then compared with the hESC regulator network derived from 10-bin vectors (shown in Figure 1A), and two commonly used indices (Dice's Coefficient D and Jaccard index J) are computed to quantify the network similarities. Here, we use A to denote the consensus network derived by using 10-bin vectors (Figure 1A) and B to denote the network derived by using different number of bins. Results comparing A and B are shown in Table S3.

It is easy to see from the table that the consensus networks derived from different resolution of tag count profiles are in general highly similar to each other, which convincingly demonstrates the robustness of the proposed network inference algorithm against different binning resolution of tag profiles. In particular, the network from 10-bin resolution has the largest number of edges, hence the highest coverage (Figure 1A) although all the other four networks overlap well with it. Therefore this network is presented in the manuscript.

Also note that increasing the resolution of the tag profiles does not necessarily lead to better network inference performance. This is not surprising, since the fluctuation of the counts in each bin is higher if the bin-size is smaller. After the non-linear logarithmic transformation and z-score normalization, noises from different bins could not cancel from each other when computing the distances between tag profiles. Therefore, choosing an appropriate resolution is necessary for inferring reliable network structures.

10. Comparing Super k-means with alternative k-means clustering algorithms

In this section, we show the Super k-means algorithm implemented in this work (which combines the k-means++ approach for initializing the cluster centers and the Hartigan's approach for optimizing the objective function) can generate more compact clusters than the original k-means++ algorithm and the classic k-means algorithm implemented in Cluster 3.0.

First, as shown in Figure S6a, we plot the sum-of-squared Euclidean distances (SSD) for grouping the data (spam_input.txt, available from http://www.ics.uci.edu/~mlearn/databases/spambase/) contained in the k-means++ software into 50 and 100 clusters. Both of the Super k-means and the k-means++ algorithm are executed 20 times and the distributions of the resulting SSDs for each algorithm are plotted. It is clear that the Super k-means algorithm yields smaller SSDs from each data to its nearest cluster center, which solidly demonstrates its superiority over the original k-means++ algorithm.

Second, to gain a deeper understanding of the advantage of using the Super k-means algorithm for profile-based clustering, we also compared it with the ordinary k-means algorithm in Cluster 3.0 for clustering the concatenated gene-wise tag profiles into 1000 groups (for learning the hESC regulator network in Figure 1A). Again, both algorithms are executed 20 times and the SSD distributions of the clustering results are plotted (Figure S6b). From the separation of the two density curves, it is easy to see that the k-means algorithm in Cluster 3.0 is hard to achieve the compactness of clusters generated by the Super k-means algorithm, no matter how many repeats it was executed. Besides, the widths of the two SSD distributions also demonstrate better stability of the Super k-means algorithm over different runs. These results demonstrate the necessity of developing the Super k-means algorithm for the profiled-based clustering task and that the off-the-shelf implementation of the k-means algorithm does not meet our stringent requirement on the clustering quality.

11. Kernel width specification by the wise normalization approach

Specifying the kernel widths for each node of the BN involves a hard combinatorial optimization problem. Yet cross-validation approaches can be used for computing the objective function during the optimization process, the computational complexity will become an issue undoubtedly. As such, we have not yet found a way to estimate the widths of kernels precisely while keeping the tractability of the algorithm in most practical settings.

The proposed wise normalization approach is a heuristic method for kernel width specification, which yields a rough estimation of the scale for each kernel. However, we observed that the final BN learning result is not sensitive to the fine-tuning of kernel widths. For most practical setting, using the results specified by the wise normalization approach is sufficient to enable good performance of BN learning .

To demonstrate this point, we performed the data re-sampling procedure two times with decreased / increased kernel widths for all nodes in the BN to derive the consensus hESC regulator network: One scales the default kernel widths computed using the wise normalization approach by a factor of $1/\sqrt{2}$ and the other by $\sqrt{2}$. All the other experimental settings are kept unchanged as in learning the consensus network in Figure 1A.

As shown by Figure S9, compared to the hESC regulator network with default kernel widths (Figure 1A), decreasing / increasing the kernel widths only leads to minor changes of the network inferred. (One edge was missing for both of the two cases). This is a strong evidence showing that fine tuning the kernel widths estimated by the wise normalization approach often simply leads to minor changes in the resulting BN structure.

12. Generalization performance of the SeqSpider algorithm

In the study cases presented in this paper, to remove background biological noise, the weight of the complexity term is always increased in the ‘tuned’ BIC scoring function (Note 3). The inferred BN structures are therefore more unlikely to ‘over-fit’ to the

training data. To demonstrate the good generalization performance of the networks learned by the SeqSpider algorithm, we performed three experiments as detailed below:

First, we sub-sample 80% and 70% training data for each fold in the 10-fold BN learning procedure for deriving the consensus network and repeat the network stability analysis as that in Figure 1B, panel f (instead of 90%). As shown from Figure S10, the resulting network stability curves are still good despite the discrepancies of training data in different folds has become much larger (only 49% of training data are expected to overlap between two folds for the 70% case). Moreover, the consensus networks derived from the two experiments above are still highly similar to the 90% data case. Specifically, only one edge (*H3K4me3--gene expression*) was missed when 80% of the data was used to repeat the analysis, and two edges (*H3K4me3--gene expression*, *Nanog--TafII*) were missed for the 70% data case. Both the network stability curves and the resulting consensus networks suggest the inferred BN structures are robust regardless of which proportion of data is sampled and hence unlikely to have been over-fitted to the training data.

Second, we randomly divide the gene-wise profiled training data (for learning the hESC regulator network in Figure 1A) into two equal-sized disjoint (training / testing) groups and cluster the data in each group into 1000 cluster centers by the Super k-means algorithm. Then, a Bayesian network is learned from the data in the training group and we compute the un-regularized KGV scores of the BN on the training/ testing groups of data, separately (Un-regularized KGV score means that the weight of the complexity term (λ) in the scoring function is set to 0 so that we do not try to 'regularize' the Kernel Generalized Variance measure, see Supplementary Methods for more details). We repeated this procedure 10 times and obtained 10 pairs of scores, as listed in Table S4a.

Since the un-regularized KGV scores in the training group were evaluated using the BNs learned from the same data, these scores are expected to be over-fitted to their respective training sets. Conversely, un-regularized KGV scores in the testing group quantify the 'generalization' capability of these BNs on unseen data, as all pairs

of (training / testing) datasets are disjoint. Therefore, it is natural to expect the scores on the training set are much larger than the test set. However, the unpaired, one-side Student's *t*-test for the scores of the training / testing sets is not significant at all (Welch's *t*-test P=0.4733, two sample *t*-test P=0.4733), which suggests the increased average scores in the training group (due to over-fitting) is not significant when compared to the inner fluctuations of the scores within each group (due to the randomness in dividing data into the training / testing parts). As a result, it is expected that the BN structures have captured more regularity of the data than nuisance factors that may cause over-fitting. To further demonstrate that this result is not obtained by chance, we also repeat the above analysis with the regularized KGV scores ($\lambda = 3.0$, the same as learning the hESC regulator network in Figure 1A). The results are listed in Table S4b. As the case of un-regularized scores, the unpaired, one-side Student's *t*-test for the regularized scores of the training / testing sets is still not significant (Welch's *t*-test P=0.2483, two sample *t*-test P=0.2482).

A more direct evidence for quantifying the over-fitting issue of the SeqSpider algorithm is to compare the BN structures learned from disjoint datasets. To this end, we performed the third, yet the most challenging experiment: First, two BNs were learned from the 1000 cluster centers of the disjoint training and testing part of the hESC data, respectively. Then, we quantify the similarity of the two BN structures directly using the Dice's Coefficient (*D*) and the Jaccard index (*J*), defined by:

$$\text{Dice's Coefficient: } D = \frac{2|A \cap B|}{|A| + |B|}, \text{ Jaccard index: } J = \frac{|A \cap B|}{|A \cup B|}.$$

where *A*, *B* are the PDAGs (characterizing the equivalence classes) of the two BNs, $A \cap B$, $A \cup B$ are the intersection and union of *A*, *B*, respectively, and $|X|$ denotes the number of edges for graph X .

This process was repeated 10 times and the resulting consistency scores were documented in Table S5. The results clearly suggest that the BN structures inferred from disjoint datasets have good match with each other in general, which is a more direct evidence for the good out-of-sample performance of the SeqSpider algorithm.

13. Comparing the L1-RPS kernel with the cross-bin kernel based on the time-warping distance and standard bin-to-bin kernels

To compare the performance of the proposed L1-RPS kernel with an alternative cross-bin kernel for vectored data, we implemented the well-known time-warping distance [31] and using it to replace the L1-RPS distance in the definition of the kernel for vectored data. Keeping all the experimental settings for learning the hESC network in Figure 1A fixed, we perform the 10-fold data re-sampling procedure using the time-warping kernel for vectored tag profiles under three different weights of the complexity term in the scoring function to derive the consensus network ($\lambda = 2, 3, 4$). As shown by Figure S7a, the resulting three consensus networks are consistently worse than the network derived by the L1-RPS kernel. This is because many important edges in the network of the L1-RPS kernel (Figure 1A) were missed by the time-warping kernel. Specifically, the two consensus networks of the time-warping kernel for $\lambda = 3, 4$ are stringent sub-networks of the L1-RPS kernel (5 and 6 edges in Figure 1A were missed in the two networks, respectively), and the network of the time-warping kernel for $\lambda = 2$ is identical to the setting of $\lambda = 3$, except for including an additional edge (TAFII-RING1B) which has no literature support.

The results above suggest the time-warping kernel is far less sensitive than the L1-RPS kernel for capturing the intrinsic relationships in the data, even when the weight of the complexity term is tuned. This is not surprising, since the chromosomal coordinates surrounding the biological landmark TSS are uniform everywhere, either stretching or compressing these coordinates artificially to match tag profiles in the time-warping kernel approach may actually distort the chromosomal distance to TSS, thus blurring real signals in the data.

Furthermore, it can be shown that the proposed L1-RPS kernel also excels the time-warping kernel in terms of the stability of network learning. In fact, the area under curve (AUC) of the network stability curve (distinguishing/not distinguishing the directionality of edges) for the L1-RPS kernel is (0.9938/0.9938) in the 10-fold

data re-sampling procedure for deriving the consensus network (Figure 1B, panel f), whereas the AUCs of network stability curves for the time-warping kernel are (0.8788/0.9224), (0.9284/0.9284), (0.9401/0.9401) for $\lambda = 2,3,4$, respectively (Figure S7a).

Besides the comparison of the proposed L1-RPS kernel with the cross-bin kernel defined by the time-warping distance, we have also compared it with two standard bin-to-bin kernels, where the L1-RPS distance in the kernel for vectored data is replaced by the L1/L2 distance (standard bin-to-bin distances for vectors). By repeating the procedure for deriving the consensus hESC network in Figure 1A, the two consensus networks that correspond to the L1/L2 distance kernels are obtained, as shown in Figure S7b.

Like the time-warping distance kernel, both the consensus networks derived from the L1 and the L2 distance kernel are stringent sub-networks of the proposed L1-RPS kernel (shown in Figure 1A). In particular, the L1 distance kernel missed 5 (31.25%) edges and the L2 distance kernel missed 9 (56.25%) edges, respectively. Besides, the network stability curves for the two standard bin-to-bin kernels are also much lower than the L1-RPS kernel (Figure 1B, panel f). The decreased sensitivity of the L1 / L2 distance kernel is perhaps because bin-to-bin distances are not resistant to small shifts of tag positions.

From the experiments above, we can conclude that the proposed L1-RPS kernel indeed has better performance than many standard approaches to defining the kernel for deep sequencing profiles. This is why we describe its basic idea and mathematical construction at length in Supplementary methods.

14. SeqSpider reverse engineered known causal relationships in CD4+ T cells

Although the hESC regulator network learned by the SeqSpider algorithm (Figure 1A) does not contain irreversible edges, we have shown that the global partially directed acyclic graph (PDAG) structure of this network has implied a stimulation-spreading

behavior around the signal-relaying hub H3K4me3, which were not available in a pure undirected, correlation based network. Here, we demonstrate that the SeqSpider algorithm could also convincingly uncover irreversible causal regulatory relationships. Specifically, we re-analyzed the ChIP-Seq datasets from CD4+ T cells which has been used to construct a discrete Bayesian network for inferring the causal regulatory relationships [8]. The results suggest that SeqSpider can faithfully reproduce previous results, capture directed interactions and even predict some new biologically relevant relationships.

First, as we discussed in the Supplementary Methods, in the SeqSpider algorithm, sequence tag profiles of the [-1k, +1k] region flanking TSSs was used as training data to infer Bayesian networks. Here, using [-1k, +1k] was to be consistent with the region defined previously in [8]. This is in sharp contrast with the previous approach where discretized total tag counts in the same genomic regions were used for BN inference (See Supplementary Methods for more details of training data curation).

After 10-folds data re-sampling procedure, we obtain the consensus network derived from the SeqSpider algorithm (Figure S12). In particular, 19 out of the 37 edges in this network overlaps with the corresponding edges in the network previously derived based on discrete data (See Figure 3c in [8] for the discrete data based network, which has 32 edges in total) regardless of the directionality of edges. Furthermore, 17 out of the 19 overlapped edges have consistent orientation (causality).

It is clear that the two networks have a statistical significant overlap (Fisher's exact test, P=2.36E-11), and the consensus of the directionality of edges is also statistically significant (Binomial test, P=3.64E-04), which solidly demonstrates that SeqSpider could faithfully reverse engineer irreversible causal interactions (A theoretical proof is given in Note 7).

Not surprisingly, we found that some of the SeqSpider-inferred network structures that are inconsistent with Yu et al. could be more biologically meaningful. As an example, for the sub-network that involves factors that determines PolII binding and gene expression, among the new interactions inferred by SeqSpider, 1)

H3K79me2 (which marks the elongation of transcription) and PolII binding collectively determine gene expression; 2) H3K4me3 and H3K27me1 jointly decide the presence of PolII binding.

These new interactions inferred by SeqSpider are in fact well-supported by literature: It has been shown that the loss of H3K27me1 is essential to initiate gene transcription in mammals [43]. Therefore, edge H3K27me1 $\rightarrow$ PolII in the SeqSpider network is biologically valid. Also, the edge H3K79me2 $\rightarrow$ Gene Exp is also demonstrated by Guenther *et al*. [40].

15. The hESC regulator network learned from an extended and most up-to-date set of deep sequencing data

The deep sequencing datasets analyzed in this work were downloaded from Gene Expression Omnibus (GEO) in May, 2010, where there were only 7 H1 cell histone marks available at that time. Now more histone mark ChIP-seq datasets are available from those GEO series. Since covalent modifications on the long tails of Histone 3, 4 have major regulatory roles, we added the 5 more H1 cell histone 3 marks (H3K4ac, H3K4me2, H3K56ac, H3K79me1, H3K79me2) from GEO to the training set and reconstruct the hESC regulator network on the currently most comprehensive datasets using the SeqSpider algorithm.

In Figure S13a, we show the three consensus networks learned from 1000, 1500 and 2000 cluster centers from the newly combined dataset and the corresponding network stability curves.

From the (directed / undirected edge) network stability curves, the network derived from 1000 /1500 cluster centers are better than the network derived from 2000 cluster centers (AUCs of the directed/undirected edge network stability curves are: (0.9437/0.9513), (0.9113/0.9606), (0.8271/0.9428) for the three networks derived from 1000, 1500 and 2000 clusters, respectively). A closer examination at the network structures suggests the network derived from 1500 cluster centers was the best among the three, since the number of edges in this network is also the largest among the three

networks, hence the highest coverage.

The network also identified more interactions than the original hESC regulator network (Figure 1A) when H3K4me2 was introduced: As an intermediate state between H3K4me1 and H3K4me3, we observed H3K4me2 is connected to both H3K4me1 and H3K4me3 in the new network. Moreover, the connection between H3K4me3 and H3K9ac in the original hESC regulator network can now be further elucidated as mediated by H3K4me2, which has been reported to be directly associated with histone 3/4 acetylation [44].

Moreover, the predicted relationship TAFII $\rightarrow$ MYC in the new network is consistent with the major role of MYC in Pol II pause release rather than Pol II recruitment [39]. Not surprisingly, some interactions in the new network (e.g., the group among H3K56ac/H3K4ac/H3K79me1/H3K79me2) are yet currently unknown.

To thoroughly compare the network derived from updated deep sequencing data (Figure S13a, k=1500) with the original hESC network (Figure 1A), we first removed the 5 new added nodes in the updated network and obtained a 'reduced network', shown in Figure S13b.

Not considering the directionality of edges, 11 of the 13 edges in this network overlaps with edges of the hESC regulator network in Figure 1A, which is highly statistically significant (P = 6.49e-11, Fisher's exact test). And both of the two new edges in this network (H3K4me1->TAFII and TAFII->MYC) are experimentally validated [39, 45]. Furthermore, all the directed edges in the 'reduced network' are oriented away from the network center H3K4me3, which is consistent with the causal interpretation of the hESC network in Figure 1A (Once H3K4me3 receives a stimulus, the influence will spread out along all the other branches in the network, See Figure S17).

As such, both the skeletons and the edge directions of this reduced network are highly consistent with the hESC network in Figure 1A. The only thing that we have to explain is why the 5 edges in Figure 1A were missed in the reduced network. These edges are:

NANOG-SOX2

NANOG-P300

NANOG-TAFII

H3K4me3-mCGLevel

H3K4me3-MYC

We checked the 10 PDAGs in the data re-sampling procedure for deriving the consensus network from updated deep sequencing data (k=1500). In fact, edges NANOG-SOX2 and NANOG-P300 appear in all the 10 runs. There are simply some ambiguities about their directionality: None of a fixed orientation for any of the two edges appears >=7 times. Since we used a relatively stringent criterion for deriving the consensus network, these two edges did not appear finally. This causality ambiguity is not due to the lack of discriminative power of SeqSpider algorithm, but simply because the circulating interaction among the three nodes NANOG, SOX2 and P300 violates the acyclicity assumption of BNs.

For the other 3 missed edges, regardless of the edge direction, both the interaction NANOG-TAFII and H3K4me3-mCGLevel appears 3 times in the 10 runs. The signal still exists but simply weaker. And the stronger new interaction TAFII->MYC replaced the old one H3K4me3-MYC.

In summary, when a less stringent criterion is used for deriving the consensus network (e.g., not considering the consistence of edge orientation), the two network could have even more overlapped edges. In statistics, learning BNs from data is a very challenging model selection problem, where the number of candidate models (BN structures) is super-exponential to the number of nodes. Obtaining a very high level statistical consistency (in our case P = 6.49e-11) is sufficient to demonstrate the robustness of the network inference algorithm.

Another concern about the hESC network derived from updated deep sequencing data (k=1500) is that H3K4me3 now connects with only 4 nodes. Therefore, its role as a central hub in the network structure is challenged. However, as shown below, this does not contradict with our observations in the original hESC network (Figure 1A).

In the network with updated deep sequencing data (k=1500), some of the connections of H3K4me3 are now mediated by H3K4me2. When we merge the two

nodes, it is clear that H3K4me3 actually connects with 5 nodes, as shown in the 'reduced network' (Figure S13b). In a broad sense, one can still claim that H3K4 methylations serve as the hub for connecting major hESC regulatory domains. This is essentially in accordance with the hESC regulator network in Figure 1A.

Second, although H3K79me1 also connect with four edges in the network with updated deep sequencing data (k=1500), it is clear that H3K79me1 and three of its interactors, H3K79me2, H3K4ac and H3K56ac form a tightly connected cluster (Note that BN only prohibit directed loops but allow cyclic structure in the undirected skeleton, it is therefore possible for the consensus network to have a densely connected component). On the other hand, the four edges of H3K4me3 spread out and link with more diverse regulatory domains in a star configuration. Therefore H3K79me1 is unlikely a competitor of H3K4me3 as hub of the network.

16. Regulatory relationships within different groups of promoters in hESC

Although we have inferred a holistic model of the regulatory relationships for some biological factors in hESC (Figure 1A), it is still interesting to further investigate the diverse behaviors of different promoters to gain a deeper understanding of the molecular interactions in the hESC regulator network. To this end, we performed the following experiments:

First, the 1000 cluster centers of the concatenated tag profiles (generated by the Super k-means algorithm) were further clustered by the hierarchical clustering algorithm in Cluster 3.0 (using Euclidean distance). After careful examination of the clustering results, five distinct groups of promoters were identified (Figure S14a).

Second, a consensus network is inferred separately for each distinct group of promoters using the 10-fold data re-sampling scheme. Here, note that the number of cluster centers in each group is very limited, which was not sufficient for learning a non-trivial BN structure. Therefore, all the gene-wise profiles that correspond to the cluster centers in each group were used as training cases. The resulting networks and

the associated network stability curves are shown in Figure S14b.

Since SeqSpider identifies potential interactions based on co-variations in training data (Figure S21), it is reasonable to expect the accuracy and stability of the networks learned for each group were not as good as the network based on all the available data (Figure 1A), as variation across the training cases in each group is even smaller than random training samples of the same size. Indeed, as indicated by the corresponding curves, stabilities of the networks inferred from Group 1, 2 were not fully satisfactory. However, it is still illuminating to examine the group-specific regulatory information conveyed in these networks.

First, consistent with the enrichment of active patterns of chromatin modification in Groups 1, 2, 3, H3K4me3 serves as a hub in the three networks, which firmly supports our hypothesis that H3K4me3 coordinates signal propagation between different regulatory domains in the original full hESC regulator network (Figure 1A). Besides, as an active histone mark, it is reasonable to see that H3K4me3 did not play such a role in the networks of Groups 4, 5, which mainly contain down-regulated chromatin patterns.

Second, the centered role of H3K27me3 in Group 5's network is also worth noting. Contrary with H3K4me3, H3K27me3 is a major repressive epigenetic mark, but here Group 5 also contains the most repressive chromatin patterns. It would be interesting to see whether H3K27me3's role in the deep repressive chromatin context is similar to H3K4me3's role in general and specifically in active promoters.

Though the regulatory relationship learned from the five gene groups seems to be quite reasonable, one may still wonder why the stability of Cluster 2's network is not satisfactory, since this group contains the largest number of chromatin patterns among the five groups.

To address this puzzle, we have to first note that for a unified treatment of network learning in the five groups, we did not employ the Super k-means algorithm to perform profile-based clustering, since the number of genes in four of these groups (except Cluster 2) is rather scarce. Second, as we have pinpointed above, Cluster 2 mainly contains active chromatin patterns, the variation across different training cases

is considerably smaller than the same number of random training samples. Therefore, stochastic noises not filtered by the Super k-means algorithm might have hindered the real regulatory signal within this group. Fortunately, Cluster 2 contains large enough number of genes for us to test this hypothesis.

Specifically, using exactly the same approach for learning the hESC network in Figure 1A, we first apply the Super k-means algorithm to cluster the concatenated profiles of the 10-fold gene-wise training data in Group 2 into 1000 cluster centers and then derive a consensus network from the 10 PDAGs learned on each fold of data (here 1000 cluster centers). The results are shown in Figure S14c.

Not surprising, we observed a dramatic improvement of the stability of network inference for Group 2. Partly due to this information filtering process, the number of edges in this network is also reduced (from 11 to 4), as some instable connections are pruned out. However, the three edges connecting to H3K4me3 in the network derived without the profile-based clustering step (Figure S14b, Cluster 2) are retained. And the four edges here better coincide with the hESC network in Figure 1A, further demonstrating the usefulness of the Super k-means algorithm for noise filtering (via profile-based clustering).

Although this result partly explained the instability of network inference without profile-clustering, one may still worry about the predictive power of SeqSpider algorithm, since the number of edges in this network (Figure S14c) is far smaller than Figure 1A. This concern can be fully eliminated by the study in Note 20, where we show when using the same, or even smaller number of randomly selected genes as Cluster 2 for training data, SeqSpider could still infer a network fairly close to Figure 1A. In other words, we demonstrated the lack of edges for the network from Cluster 2 is not because of the poor performance of SeqSpider on small sets of training data, but is simply due to the limited regulatory information contained in Cluster 2.

17. Inferring the mESC regulator network

To see if an independent regulatory network among the same set of TFs and epigenetic modifications from the NGS data of self-renewing mouse embryonic stem

cells (mESCs) has a similar structure as the network learned from hESCs, we downloaded the mESC ChIP-Seq data for 6 TFs (Nanog, Oct4, Sox2, Klf4, c-Myc, p300) from GSE11431. We also collected the ChIP-Seq data for another TF (TAF1), 5 histone modifications (H3K27ac, H3K27me3, H3K36me3, H3K4me1, H3K4me3), the MeDIP-Seq data (measuring genome-wide DNA methylation patterns) and the RNA-Seq data of mESCs from GSE36114.

The BED files in GSE36114 contain mapping results to the UCSC mouse genome release mm9, whereas the BED files from GSE11431 are mapping results to mm8, we therefore remapped the raw sequencing tags of the GSE11431 dataset to mm9 using Bowtie [46] (command line: bowtie -p 20 -S -q -k 1 -m 1 -v 2 -y --best index/mm9, allowing 2 mismatches). The percentage of tags that mapped to mm9 (mapping ratio) is generally acceptable for 5 of the 6 TFs except NANOG, for which the mapping ratio is only 0.84%. So, for NANOG, we further trimmed the reads 16 bp from 3' end before alignment to increase the mapping ratio to 1.69%. (We have double-checked the map of NANOG ChIP-Seq data to mm9 using SOAP2 [47]. The mapping ratio is still low (1.34%), which confirmed the results obtained using Bowtie [46].)

To infer the mESC regulator network, we employ the SeqSpider software (with default setting) to scan the tag distribution signals in the TSS [-2kb, +2kb] regions for each ChIP-Seq / MeDIP-Seq data. After this step, tag distributions around TSSs are represented by 10-dimensional vectors, instantiated by the numbers of sequence tags that mapped to 10 disjoint, equal-sized (0.4kb) bins in the TSS [-2kb, +2kb] region. As in [48], the mESC RNA-Seq data is processed using Bowtie [46] to obtain the FPKM values for each gene (represented by a real-valued variable). Similar to inferring the hESC network, we applied the logarithm transformation ($y = \log_2(x + c)$, c is a positive constant) and z-score normalization (to standardize the data to zero mean and unit variance) to each bin of the vectors and gene expression value (before sampling 10 subsets of training data for learning the consensus network). Here, we set c to the default value ($c = 1$) for ChIP-Seq / MeDIP-Seq data sets to avoid zero tag count in the vectors and tested different values of c ($c = 1, 0.1, 0.01, 1e-3, 1e-6$) to

normalize the FPKM value of gene expression for the best reproducibility and coverage of the consensus network.

We adopt exactly the same Bayesian network learning strategy used for inferring the hESC network to derive the mESC consensus network.

To make a thorough comparison with the hESC regulator network in Figure 1A, we learn mESC networks under two slightly different settings to explore the key molecular interactions / regulations at the genes' promoter regions of mESC, where H3K36me3, preferentially enriched in TTS, is either not included or its TTS signals are included in the analysis.

We first learn a 13-nodes mESC network using SeqSpider, only considering the tag signals at the TSS [-2kb, +2kb] regions. We find that the network inference results are generally stable across different values of c for normalizing the FPKM value. In particular, the network inferred from $c = 0.01$ has the most number of edges and is presented in Figure S25A for its comprehensiveness (Comparing with it, the networks inferred under $c = 1e-3$ or $c = 1e-6$ only missed one edge, which demonstrates the robustness of this network inference result). We can see 5 out of the 10 edges in the mESC network overlap with the edges or feedback edges in the hESC network (see Figure S20), which is statistically significant (P=0.019, Fisher's exact test). They are H3K4me3-DNA methylation, H3K4me3-gene expression, H3K4me3-Myc, Klf4-Oct4, P300-Sox2. Among the remaining 5 edges, the interactions Oct4-Sox2, Oct4-Myc are well-known in ESCs and the edge H3K27ac-H3K4me3 exemplifies the communication between enhancers and promoters. In summary, these results demonstrate the robustness of the predicted mESC network and its high quality. Moreover, the significant overlap and the non-negligible difference between the mESC / hESC network are consistent with the recent finding of cross species epigenome comparison [48]: there are both significant similarities and clear differences between the human and mouse epigenomes.

In particular, as in the hESC network, we found H3K4me3 still serves as the hub in the mESC network. It directly connects to four nodes: DNA methylation, H3K27ac, Myc and gene expression. Except gene expression, each of the three nodes represents

a distinct regulatory domain (DNA methylation as an important repressive mark, H3K27ac stands for enhancer signals, Myc conveys information of several key ESC TFs for transcription control [39]), which is consistent with the role of H3K4me3 in hESC for propagating information among different regulatory domains. Still, there are some differences for edges connecting H3K4me3 in the two networks. For example, the edge H3K4me3-Oct4 in hESC network has been demonstrated to be true [42], but it did not appear in the mESC network. It would be interesting to experimentally test whether this interaction is specific to human ESCs.

We also try to learn mESC networks using the TSS [-2kb, +2kb] signal for the 13 nodes above and the TTS [-2kb, +2kb] signal for H3K36me3, just as learning the hESC network. The most robust mESC network learned from this scenario is shown in Figure S25B ($c = 1.0$). Similar to previous results, the network structure is stable under similar parameter setting (with only one superfluous edge for $c = 0.1$, but the network structure becomes less stable for $c = 0.01$ and smaller c values).

When comparing this mESC network with previous ones that only focus on interactions at the genes' promoter regions, we find only one difference: Now gene expression is connected to H3K36me3 rather than H3K4me3. This new edge is also biologically correct and the larger c value used in network inference reflects the fact that H3K36me3 is more correlated with gene expression at higher expression levels.

Interestingly, when we apply the post-BN feedback edge search algorithm (see Supplementary Methods) to this scenario, the edge H3K4me3-gene expression is recovered with high confidence. Specifically, we execute this algorithm 50 times on the 10-fold training data of this 14-nodes mESC network (5 times on the data of each fold to account for the stochasticity in graph search) and count the times of occurrence for each reported feedback edge in the 50 runs. We find that H3K4me3-gene expression is the only edge that appears in more than 50% of the runs (26 times). The reason that this edge did not appear in the consensus network is perhaps because H3K36me3's correlation with gene expression is stronger, further adding the edge H3K4me3-gene expression will create a loop in the BN structure through H3K36me3's unstable connection with other modifications. By considering this

feedback edge, the overlap between the 14-nodes mESC network with the hESC network in Figure S20 becomes more statistically significant (P=0.0032, Fisher's exact test).

To conclude, in this section, we have inferred a high quality mESC regulator network from heterogeneous NGS datasets using SeqSpider. The global network structure is generally stable under different network learning contexts. The mESC network has strong overlap yet also non-negligible differences with the hESC network, showing species-specific strength difference of molecular interactions. Nevertheless, H3K4me3's role as the signal propagation hub connecting diverse regulatory domains is clear in both of the two ESC networks.

18. Analysis of known regulatory relationships not appeared in the network

A few known regulatory relationships, such as Nanog-Oct4, Sox2-Oct4, P300-Sox2 P300-Oct4, were not predicted by the hESC regulator network shown in Figure 1A. It is interesting to know why these interactions were not included in the network. To this end, we computed the (regularized) kernel generalized variance score for each of the four edges above (the weight parameter in the BN scoring function is set the same as for learning the hESC regulator network ($\lambda = 3.0$)), shown in Table S6a.

For comparison purpose, we also compute the regularized KGV scores for the 16 edges in the hESC regulator network (with the same setting of the weight parameter ($\lambda = 3.0$)), shown in Table S6b.

It is easy to see that except the interaction P300-SOX2 has a positive and relatively larger regularized KGV score (308.842062), the scores for all the other three missed edges are negative and consistently smaller than the smallest score of the edges in the network (20.739222), which implies that these known interactions were not well manifested in the ChIP-seq data of the particular cell context. This could be due to the fact that the ChIP-seq data reflects only the steady state of the cells, while the four known interactions are more dynamic and context dependent. Given more

dynamic data, such as ChIP-seq experiments during the course of hESC self-renewal and differentiation, some of these interactions might be more discernable.

The lack of the edge P300-SOX2 could be explained by the fact that we already have two edges in the hESC regulator network (SOX2-Nanog, Nanog-P300), and hence the addition of P300-SOX2 may create a loop in the network structure (which is prohibited by the acyclicity constraint of BNs). In fact, if we apply the post-BN learning graph search strategy to search for feedback edges that were missed in the BN structure, the interaction P300-SOX2 can be recovered as expected. Moreover, this approach also retrieved the edge OCT4-TAFII, indirectly connecting OCT4 to NANOG (See Note 19 for more details).

We also perform Student's *t*-test between the two groups of regularized KGV scores, one for 16 edges in the hESC network and the other for the 4 edges missed in the network. The single ended p-value is statistically significant (Welch's *t*-test P=0.03078, two sample *t*-test P=0.0002386), which suggests that these missed edges indeed have significantly less data support than edges in the network. Moreover, if we remove P300-SOX2 from the 4 missing edges (as it can be recovered as a feedback edge) and re-perform the Welch's / two sample *t*-test between the two groups (3 missing edges vs. 16 edges in the hESC network), the single ended p-values become even more significant (Welch's *t*-test P=6.054e-11, two sample *t*-test P=1.649e-6) .

19. Post-BN learning graph search strategy successfully retrieves biologically relevant feedback edges

We described the post-BN learning graph search strategy in Supplementary Methods. Here, we show although this improvement does not consider self and reciprocal feedback loops (such as A<->A or A<->B) as mentioned above, it did discover many interesting biological interactions from the hESC dataset.

Specifically, we run this post-BN graph search procedure 50 times on the 10-fold sub-sampled datasets for learning the consensus hESC network in Figure 1A. Four edges that frequently appear in the 50 runs (with >10% occurrence rate) are identified,

as listed in Table S9. See also Figure S20 for the contexts of the four feedback edges on the hESC regulator network.

It is easy to see that (a) the number of feedback edges detected for each BN (1.7 on average) is much smaller than the number of edges in the BN itself; (b) the strength of feedback edges are typically much weaker than the edges in the BN, since all edges in the consensus network must appear equal or more than 7 times in the 10 PDAGs within the data re-sampling based network learning procedure, but even the highest occurrence frequency for feedback edges is often much smaller (in this case 40%). Due to these two facts, we believe that the main causal effects identified from the BN structure are basically reserved and the learned feedbacks are simply meaningful supplements to the main effects. In this way, we partly overcome the acyclic constraint of BNs while not adding too much confusion to the interpretation the network structure.

Literature mining suggests all the four feedback interactions are biologically relevant. First, it has been experimentally demonstrated that the OTC4 to TAFII interaction was physically mediated by Esrrb [49]. Second, it has been show that P300 acetylates SOX2's DNA binding domain and therefore increasing the global acetylation level of ES cells [50]. Finally, it is well known that both H3K4me1 and H3K27ac marks are specifically enriched in active enhancers, and that enhancers have to first interact with the TFIID complex to recruit PolII to initiate transcription [45]. Therefore, the interaction of the two marks with TAFII is representative of how active enhancer triggers transcription.

Finally, we would like to note that through the explicit search for feedback edges, some known regulatory relationships not represented in the hESC regulator network are perfectly recovered. For example, the retrieved feedback interaction P300-SOX2 exactly completes the circuitry between NANOG, SOX2 and P300. The enhancer's role for initiate gene transcription is well depicted by H3K4me1-TAFII and H3K27ac -TAFII. Last but most importantly, the TAFII-OCT4 interaction discovered in the feedback edge hunting stage eventually connects the three key hESC transcription factors NANOG, OCT4 and SOX2 together (Figure S20):

OCT4-TAFII-NANOG-SOX2

That is, OCT4 and NANOG collaborate in the regulation of the transcription initialization complex; NANOG and SOX2 directly interact with each other. Unfortunately, the interaction between OCT4 and SOX2 was still not inferred from this dataset, as the regularized KGV score for this interaction is far below zero (-269.076411). Perhaps this interaction is better manifested in other cellular contexts.

To conclude, with this innovative technique to break up the acyclic constraint in BNs, we have greatly improved the quality of network inference by recovering more biologically relevant information. In this way, the potential of SeqSpider algorithm is further demonstrated. Albeit the network structure still has some imperfectness, it is unrealistic to assume that a data mining tool would perfectly reconstruct all the biological knowledge given the limited static information in the training data.

20. The robustness of SeqSpider algorithm on small training samples

We have investigated the generalization performance of the SeqSpider algorithm in Note 12, where the key focus is to compare the performance of the algorithm on the training and test datasets. In this section, we study a related but different question: whether SeqSpider could convincingly reproduce the hESC network in Figure 1A using small training samples?

Before answering this question, remember that we have repeated the approach for learning the hESC network in Figure 1A to the Cluster 2 gene group in Figure S14a, and obtained a network with only 4 edges (shown in Figure S14c). It is interesting to know whether this is simply because Cluster 2 contains too few training samples.

To this end, we randomly selected 10024 (41.76%) genes (the same number as Cluster 2) without replacement from the genome and employing the Super k-means algorithm to group the concatenated deep sequencing profiles of these genes into 1000 cluster centers (i.e., profile-based clustering). Then, we learn a BN from the 1000 cluster centers using the default parameter setting of SeqSpider (as inferring the network in Figure 1A) and convert it into a PDAG. The above process is repeated 10

times and finally (i) we compare the similarity of the 10 PDAGs with the consensus hESC network in Figure 1A (quantified by the Dice's coefficient and Jaccard index); (ii) we count the degree of H3K4me3 in the 10 networks. The results are summarized in Table S10a. We can see that (i) all the 10 networks learned on small training samples are highly similar to the hESC regulator network in Figure 1A and (ii) the degrees of H3K4me3 in the 10 networks basically keep the constant level as in Figure 1A, which demonstrates the stability of its role as an information propagation hub.

To further prove this good result is not accidental, we repeat the above analysis using even smaller number (40%) of randomly selected genes. Again, qualitatively the same results are obtained as expected, as shown in Table S10b.

From the experiments above, we have clearly demonstrated the good stability of the SeqSpider algorithm on small training samples and proved the failure of learning an informative network from Cluster 2 gene group (Figure S14c) is simply due to the lack of enough data variation within this group. Together with the extensive experiments in Note 9, 11, 12 that also evaluate the stability issue from different perspectives, the robustness of SeqSpider algorithm is solidly demonstrated.

21. Deciphering enhancer-TSS interactions, a potential application of SeqSpider

In this work, we constructed the hESC regulator network based on the TF/ epigenetic signal at the TSS [-2kb, +2kb] (or TTS -/+2kb for H3K36me3) region. In the future, using the same technique, it is possible to infer a separate regulatory network at the enhancer sites when precise, genome-wide maps of enhancers are available.

Note that one should not feel confused if the network learned from the enhancer sites is rather different from or at least not identical to the network derived from the TSS sites. This is because some transcription factors and epigenomic modifications preferentially enrich at the TSS sites while others for enhancers [32, 51]. For example, the H3K4me3 signal is almost exclusively at TSS, and absence at enhancer sites and the H3K36me3 signal only preferentially enriches at the gene body region and TTS

sites, but not at the enhancer sites. On the contrary, H3K4me1 and H3K27ac are exclusively enriched on enhancers. Since the node sets for the two networks could be quite different, the two networks are not exactly comparable.

Although learning a regulatory network at the enhancer site is meaningful by itself. A more biologically relevant application of SeqSpider is probably to infer long distance interactions between TFs and epigenetic modifications at enhancers and TSSs, when more such enhancer-TSS interaction data become available. It is also worth noting that the presence of sequence motifs and the interaction of motif pairs at the genomic regions used for network inference are good evidences for validating the accuracy and biological relevance of the network, since the motif-motif interaction data is purely genomic sequence based, which is independent of the deep sequencing data used for network inference. Therefore, as the present study case of the TSS flanking regions, motif interaction data can be also derived from enhancers to validate the hESC network inferred from enhancer regions.

Finally, we would like to point out that since many enhancers are enclosed in the vicinity of TSS sites studied in this work, our hESC network learned from TSSs has already captured some interaction between enhancer marks and the transcriptional machinery. Indeed, with the proposed post BN-learning graph search algorithm for finding 'feedback edges' (Supplementary Methods, Note 19), two known interactions between enhancers and TSSs are discovered: (H3K4me1—TAFII and H3K27ac—TAFII). It is hoped that by incorporating more enhancer data into network inference, more enhancer-TSS interactions could be inferred by SeqSpider.

It is worth noting that potential applications of SeqSpider are not limited to the prediction of regulatory networks. Recently, NGS technique has been used to determine the protein-protein interactions in human at a much higher accuracy than before [52]. It would be interesting to adapt SeqSpider for inferring other types of biological networks, such as protein interactomes, from NGS datasets.

22. A brief user's manual for SeqSpider

Installation of SeqSpider only requires unpacking the files in the "seqspider.zip" file

on any Linux platform and adding the directory to the PATH.

The program "SeqSpider.pl" curates training data and builds Bayesian networks of TFs/Histone modifications/DNA methylations from the raw deep sequencing reads (usually represented in the BED format files) and a REFFLAT file (indicating the position of the TSS sites of the genes, which could be downloaded from UCSC genome browser) as input. The following example uses the REFFLAT file “data/hg18/refFlat.txt” and several BED format files under the “data” directory as input. The output gene-wise training data is saved in the “test_matrix.txt” file. By using Super k-means algorithm to group the 10-fold sub-sampled training data into 1000 clusters and inferring a BN on each dataset, the final consensus network is saved in the “test_matrix.sif” file. The command line for calling “SeqSpider.pl” is:

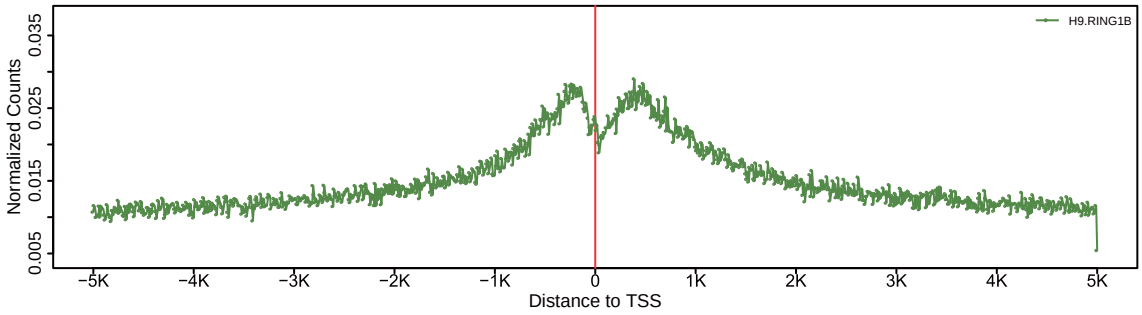
perl SeqSpider.pl --refSeq data/hg18/refFlat.txt --methyFiles “data/*.bed” --output test_matrix.txt --SKcluster 1000

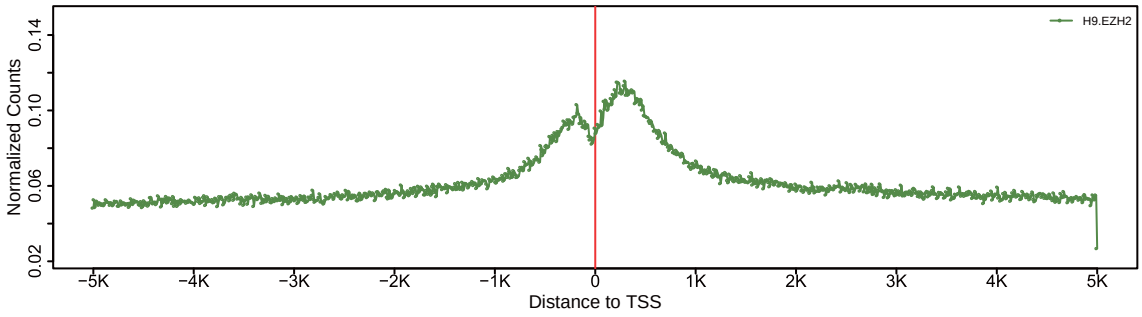
Often, a user may need to infer BNs from his/her own training data. In such cases, he/she only needs to execute "exeABCD.pl", the core module of the SeqSpider package, for building Bayesian networks from a well-curated matrix file. The matrix file is tab delimited, each row represents a gene, and each column a node (e.g. a regulator). If a node is represented by a vector (such as the 10-bin vectors used to characterize tag profiles at TSS regions for a histone modification), the header of those columns should be the same, as shown in the example file "data/human_ESC_regulaters.tsv". The following is an example of calling “exeABCD.pl”:

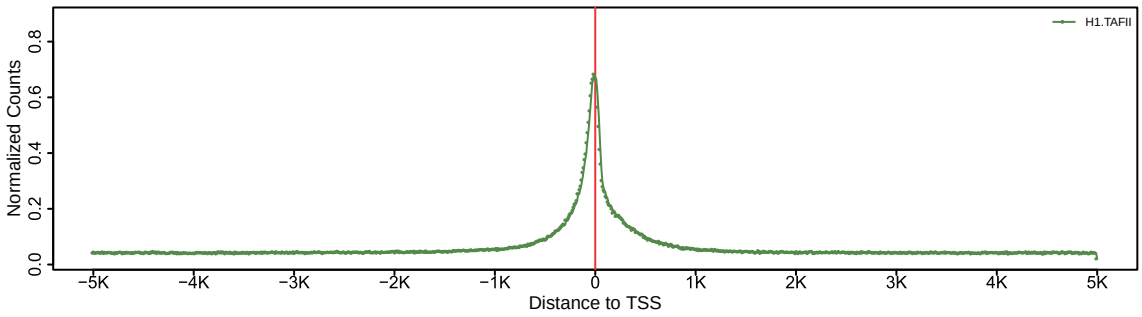
perl exeABCD.pl --input data/test_matrix.txt

Supplementary Figures

Figure S1







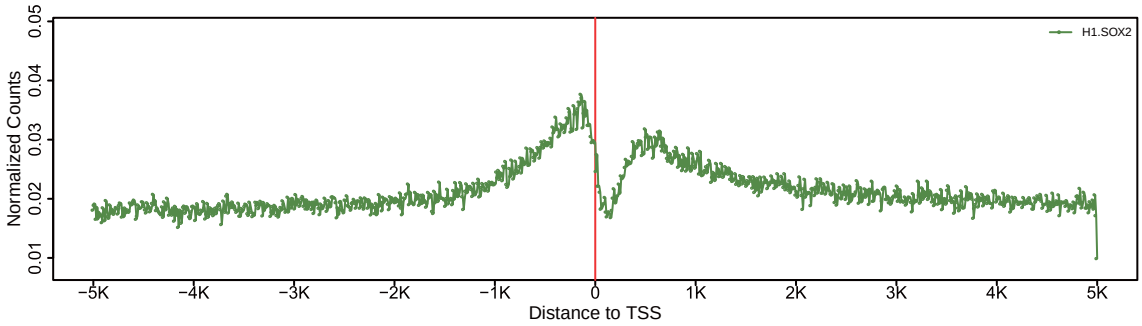
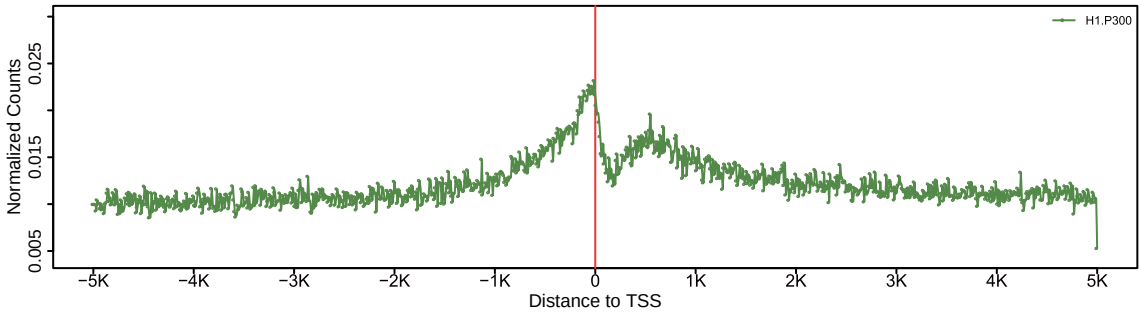
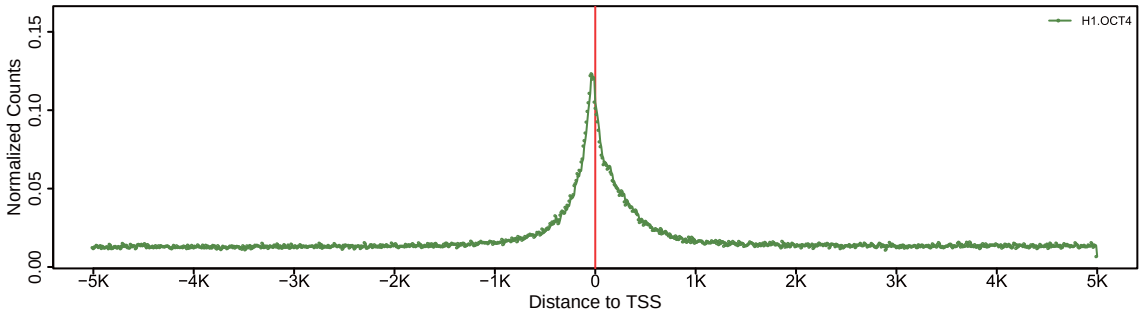
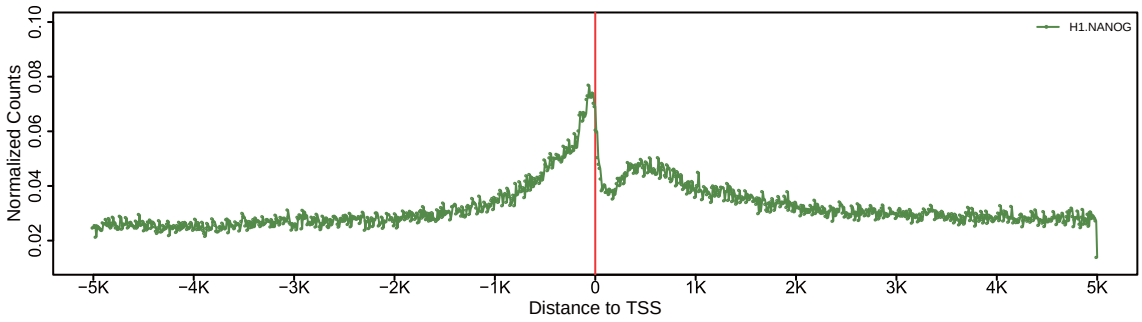


Figure S1


H1.P300
Normalized Counts
0.005
0.015
0.025
−5K
−4K
−3K
−2K
−1K
0
1K
2K
3K
4K
5K
Distance to TSS


H1.OCT4
Normalized Counts
0.00
0.05
0.10
0.15
−5K
−4K
−3K
−2K
−1K
0
1K
2K
3K
4K
5K
Distance to TSS


H1.NANOG
Normalized Counts
0.02
0.04
0.06
0.08
0.10
−5K
−4K
−3K
−2K
−1K
0
1K
2K
3K
4K
5K
Distance to TSS

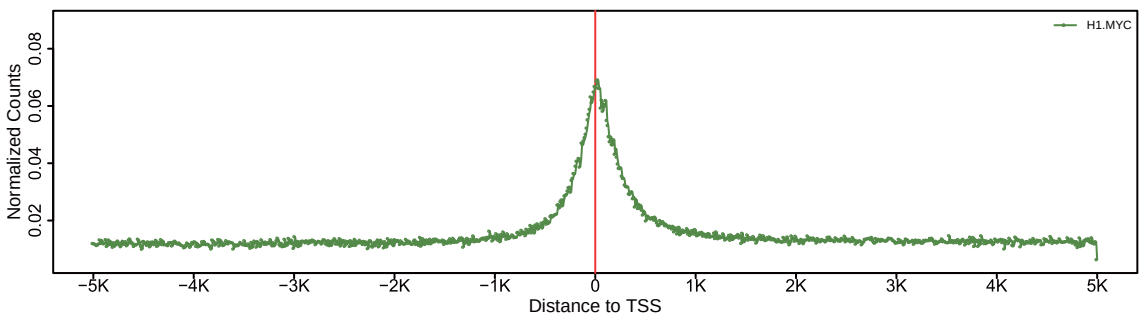
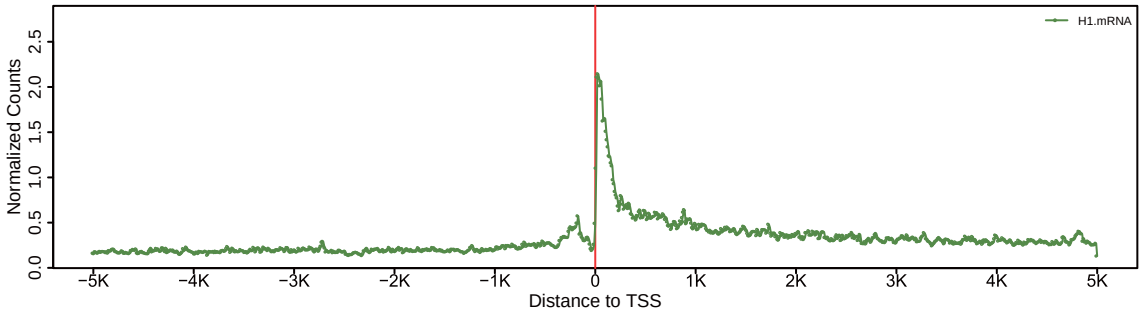
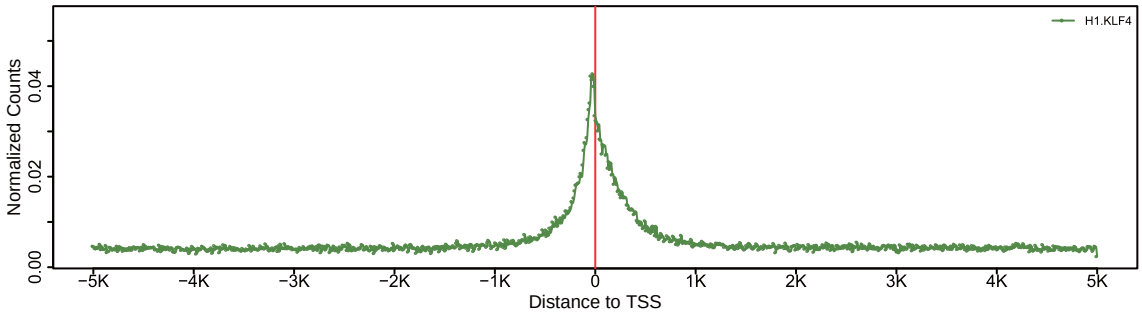
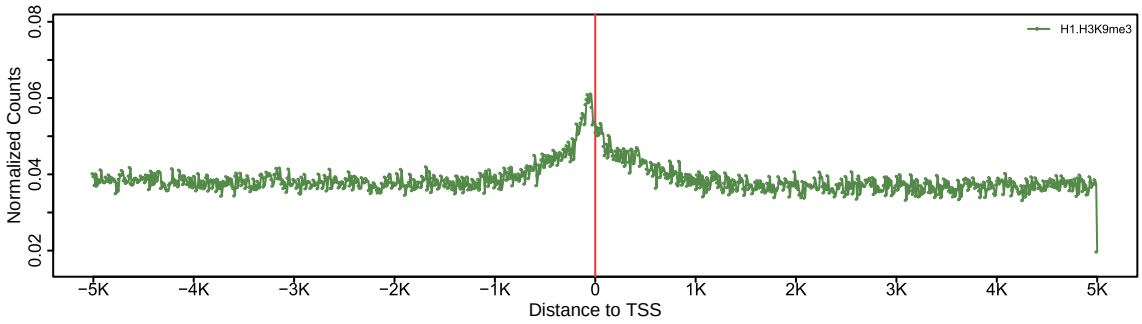

H1.MYC
Normalized Counts
0.02
0.04
0.06
0.08
−5K
−4K
−3K
−2K
−1K
0
1K
2K
3K
4K
5K
Distance to TSS

Figure S1


H1.mRNA
Normalized Counts
0.0
0.5
1.0
1.5
2.0
2.5
−5K
−4K
−3K
−2K
−1K
0
1K
2K
3K
4K
5K
Distance to TSS


H1.KLF4
Normalized Counts
0.00
0.02
0.04
−5K
−4K
−3K
−2K
−1K
0
1K
2K
3K
4K
5K
Distance to TSS


H1.H3K9me3
Normalized Counts
0.02
0.04
0.06
0.08
−5K
−4K
−3K
−2K
−1K
0
1K
2K
3K
4K
5K
Distance to TSS

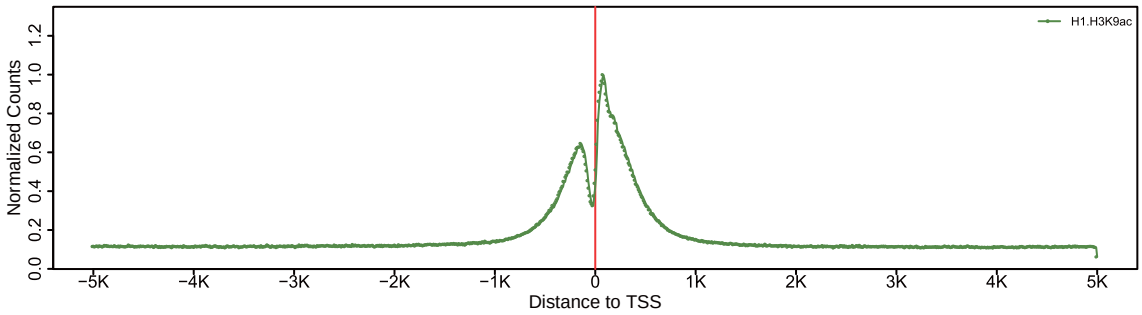
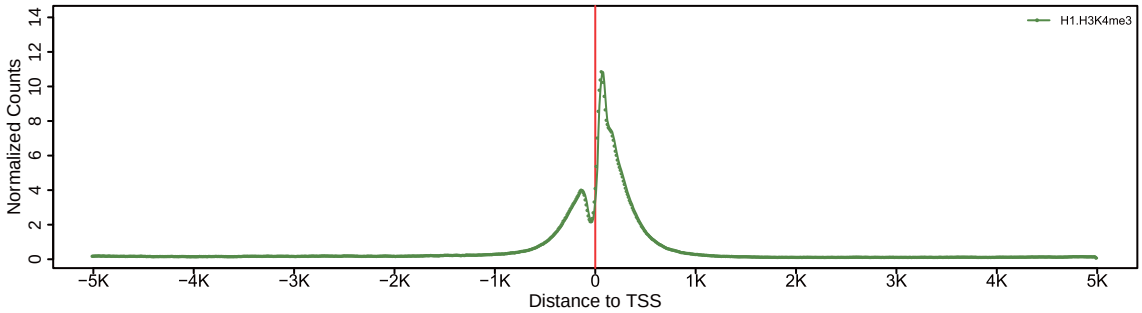
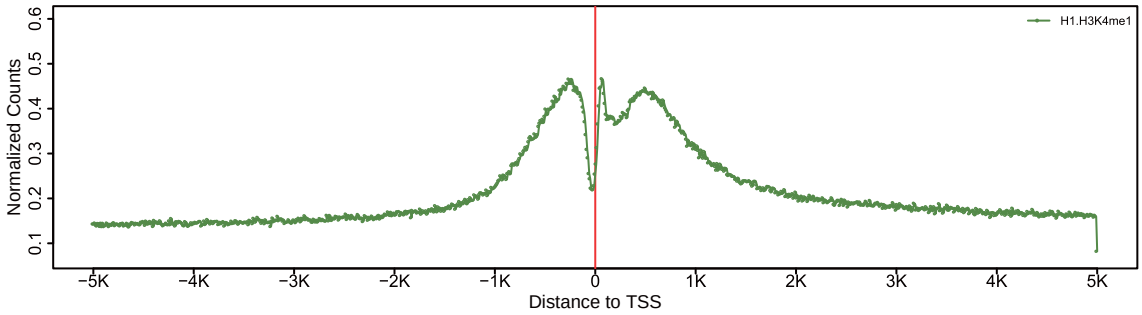
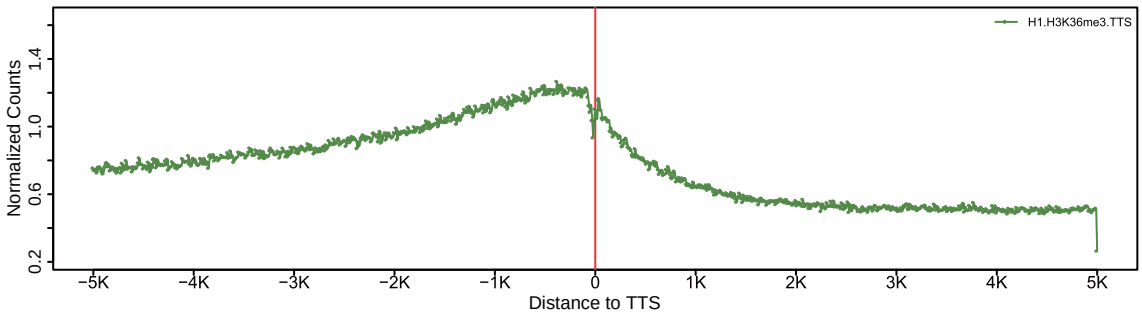

H1.H3K9ac
Normalized Counts
0.0
0.2
0.4
0.6
0.8
1.0
1.2
−5K
−4K
−3K
−2K
−1K
0
1K
2K
3K
4K
5K
Distance to TSS

Figure S1


H1.H3K4me3
Normalized Counts
0
2
4
6
8
10
12
14
−5K
−4K
−3K
−2K
−1K
0
1K
2K
3K
4K
5K
Distance to TSS


H1.H3K4me1
Normalized Counts
0.1
0.2
0.3
0.4
0.5
0.6
−5K
−4K
−3K
−2K
−1K
0
1K
2K
3K
4K
5K
Distance to TSS


H1.H3K36me3.TTS
Normalized Counts
0.2
0.6
1.0
1.4
−5K
−4K
−3K
−2K
−1K
0
1K
2K
3K
4K
5K
Distance to TTS

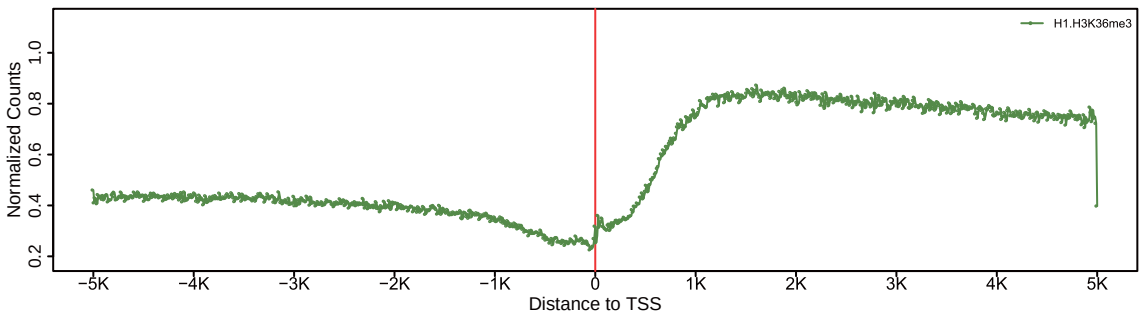
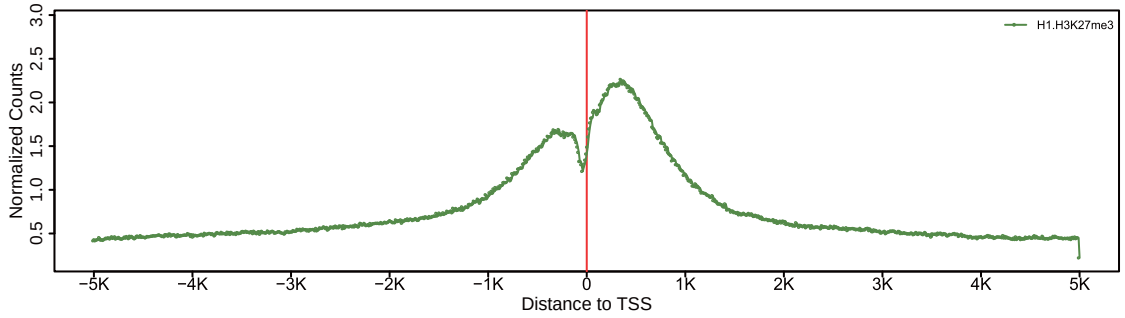
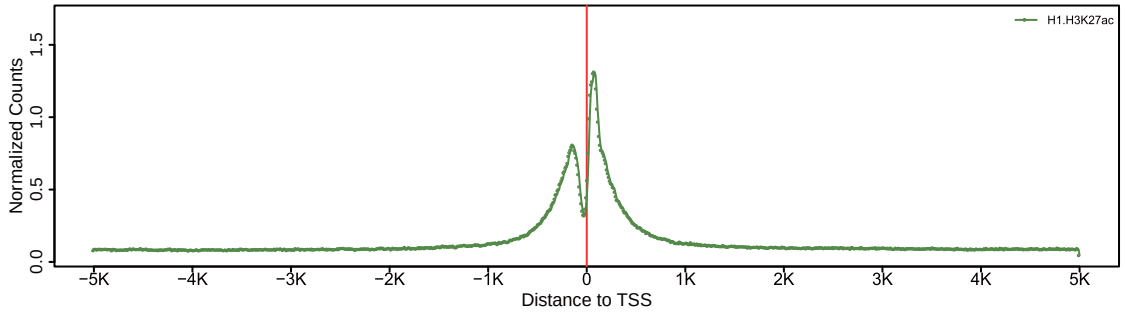
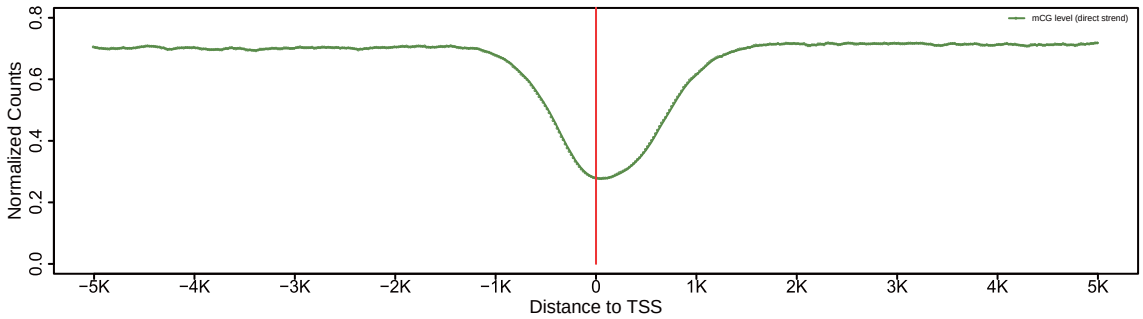

H1.H3K36me3
Normalized Counts
0.2
0.4
0.6
0.8
1.0
−5K
−4K
−3K
−2K
−1K
0
1K
2K
3K
4K
5K
Distance to TSS

Figure S1


H1.H3K27me3
Normalized Counts
0.5
1.0
1.5
2.0
2.5
3.0
−5K
−4K
−3K
−2K
−1K
0
1K
2K
3K
4K
5K
Distance to TSS


H1.H3K27ac
Normalized Counts
0.0
0.5
1.0
1.5
−5K
−4K
−3K
−2K
−1K
0
1K
2K
3K
4K
5K
Distance to TSS


mCG level (direct strend)
Normalized Counts
0.0
0.2
0.4
0.6
0.8
−5K
−4K
−3K
−2K
−1K
0
1K
2K
3K
4K
5K
Distance to TSS

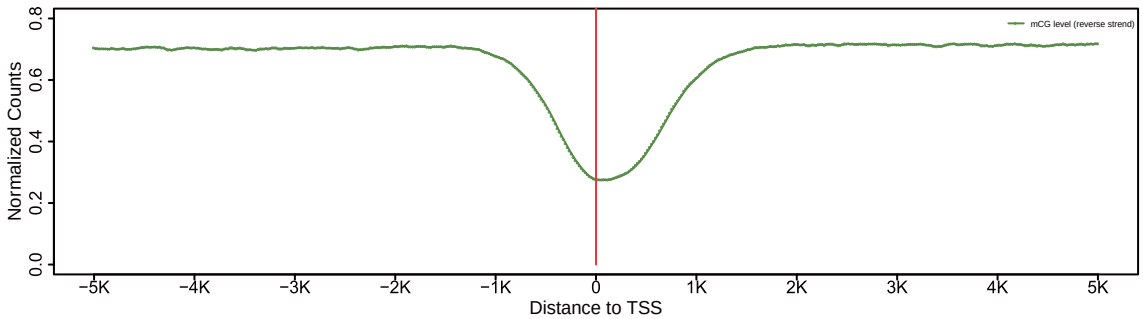

mCG level (reverse strend)
Normalized Counts
0.0
0.2
0.4
0.6
0.8
−5K
−4K
−3K
−2K
−1K
0
1K
2K
3K
4K
5K
Distance to TSS

Figure S1. The averaged sequence tag distributions in the TSS +/- 2kb region for epigenetic modifications / transcription factors. An exception is H3K36me3, whose averaged tag distribution in the preferentially enriched TTS +/- 2kb region is also plotted (See Figure S2 for the averaged tag distributions around gene body).

Figure S2

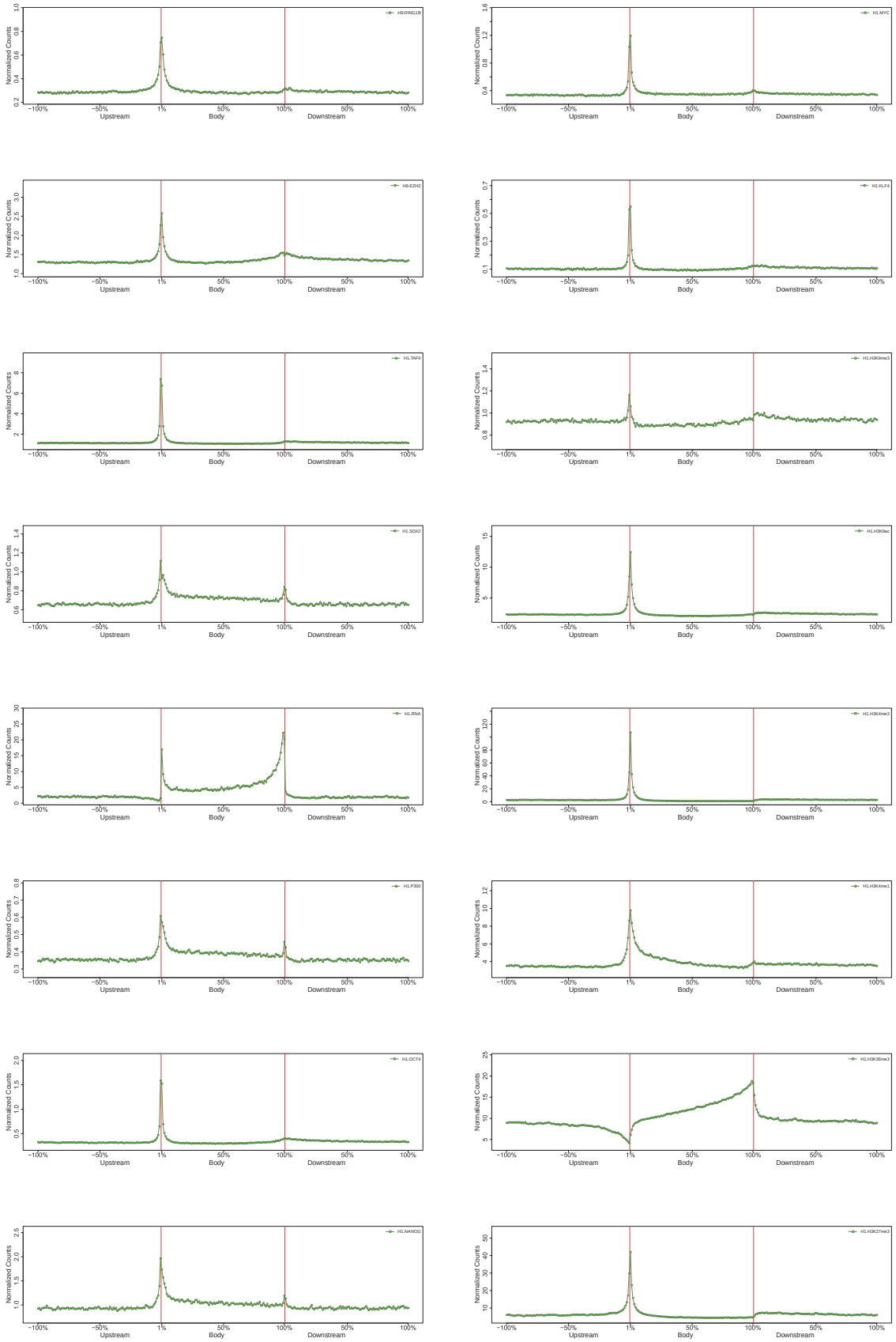

H3.RING1B
H3.MYC
H3.EZH2
H3.KLF4
H3.TAF3
H3.H3K9me3
H3.SOX2
H3.H3K9ac
H3.RNA
H3.H3K4me3
H3.P300
H3.H3K4me1
H3.OCT4
H3.H3K36me3
H3.NANOG
H3.H3K27me3
Normalized Counts
−100%
−50%
1%
50%
100%
50%
100%
Upstream
Body
Downstream

Figure S2. The averaged ChIP-seq tag distribution normalized against gene bodies. Note that different from most epigenetic modifications / transcription factors, the ChIP-seq tags for H3K36me3 are preferentially enriched around the TTS regions rather than the genes' promoter regions.

Figure S3

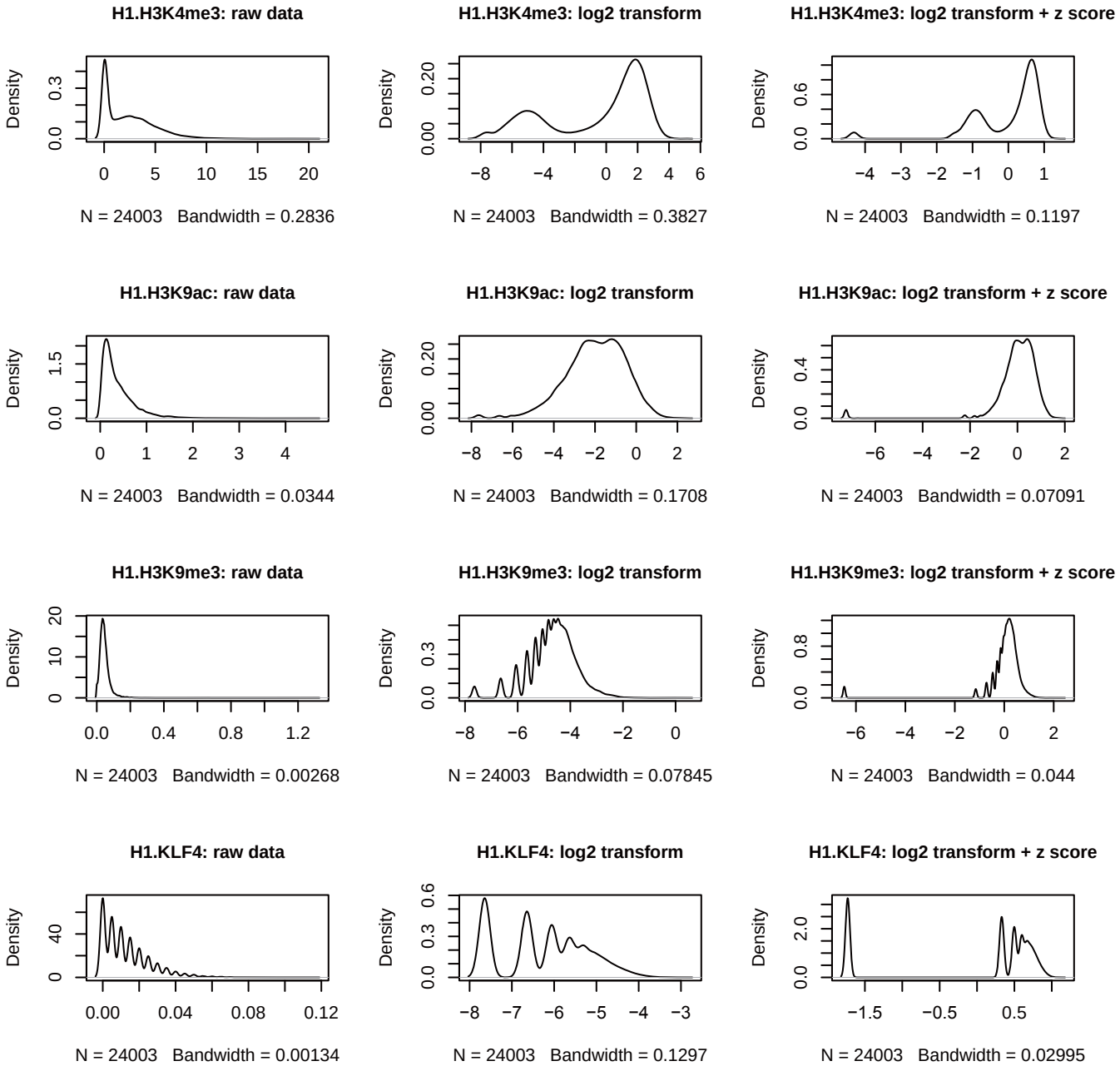

H1.H3K4me3: raw data
Density
N = 24003 Bandwidth = 0.2836
H1.H3K4me3: log2 transform
Density
N = 24003 Bandwidth = 0.3827
H1.H3K4me3: log2 transform + z score
Density
N = 24003 Bandwidth = 0.1197
H1.H3K9ac: raw data
Density
N = 24003 Bandwidth = 0.0344
H1.H3K9ac: log2 transform
Density
N = 24003 Bandwidth = 0.1708
H1.H3K9ac: log2 transform + z score
Density
N = 24003 Bandwidth = 0.07091
H1.H3K9me3: raw data
Density
N = 24003 Bandwidth = 0.00268
H1.H3K9me3: log2 transform
Density
N = 24003 Bandwidth = 0.07845
H1.H3K9me3: log2 transform + z score
Density
N = 24003 Bandwidth = 0.044
H1.KLF4: raw data
Density
N = 24003 Bandwidth = 0.00134
H1.KLF4: log2 transform
Density
N = 24003 Bandwidth = 0.1297
H1.KLF4: log2 transform + z score
Density
N = 24003 Bandwidth = 0.02995

Figure S3

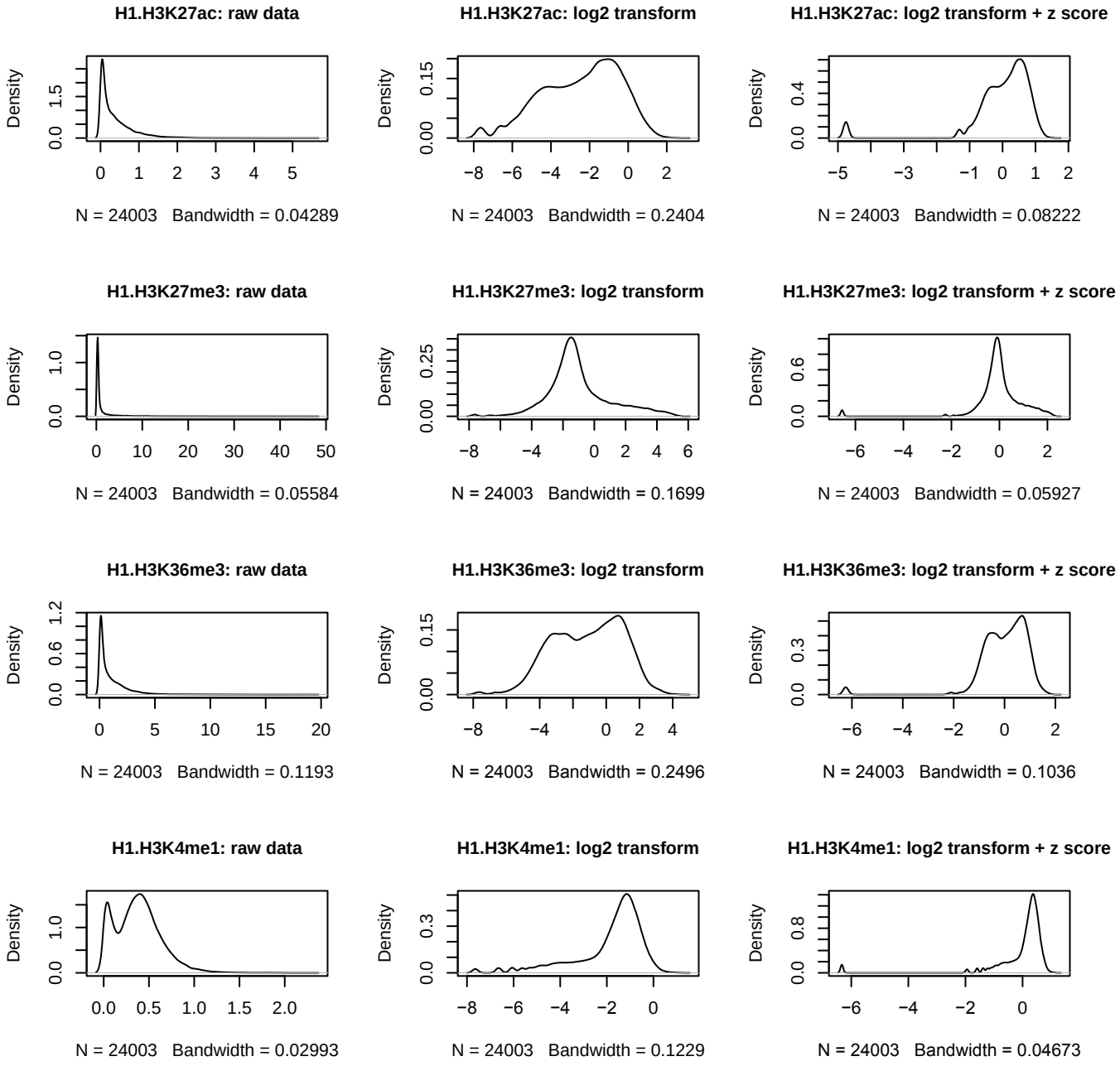

H1.H3K27ac: raw data
Density
N = 24003 Bandwidth = 0.04289
H1.H3K27ac: log2 transform
Density
N = 24003 Bandwidth = 0.2404
H1.H3K27ac: log2 transform + z score
Density
N = 24003 Bandwidth = 0.08222
H1.H3K27me3: raw data
Density
N = 24003 Bandwidth = 0.05584
H1.H3K27me3: log2 transform
Density
N = 24003 Bandwidth = 0.1699
H1.H3K27me3: log2 transform + z score
Density
N = 24003 Bandwidth = 0.05927
H1.H3K36me3: raw data
Density
N = 24003 Bandwidth = 0.1193
H1.H3K36me3: log2 transform
Density
N = 24003 Bandwidth = 0.2496
H1.H3K36me3: log2 transform + z score
Density
N = 24003 Bandwidth = 0.1036
H1.H3K4me1: raw data
Density
N = 24003 Bandwidth = 0.02993
H1.H3K4me1: log2 transform
Density
N = 24003 Bandwidth = 0.1229
H1.H3K4me1: log2 transform + z score
Density
N = 24003 Bandwidth = 0.04673

Figure S3

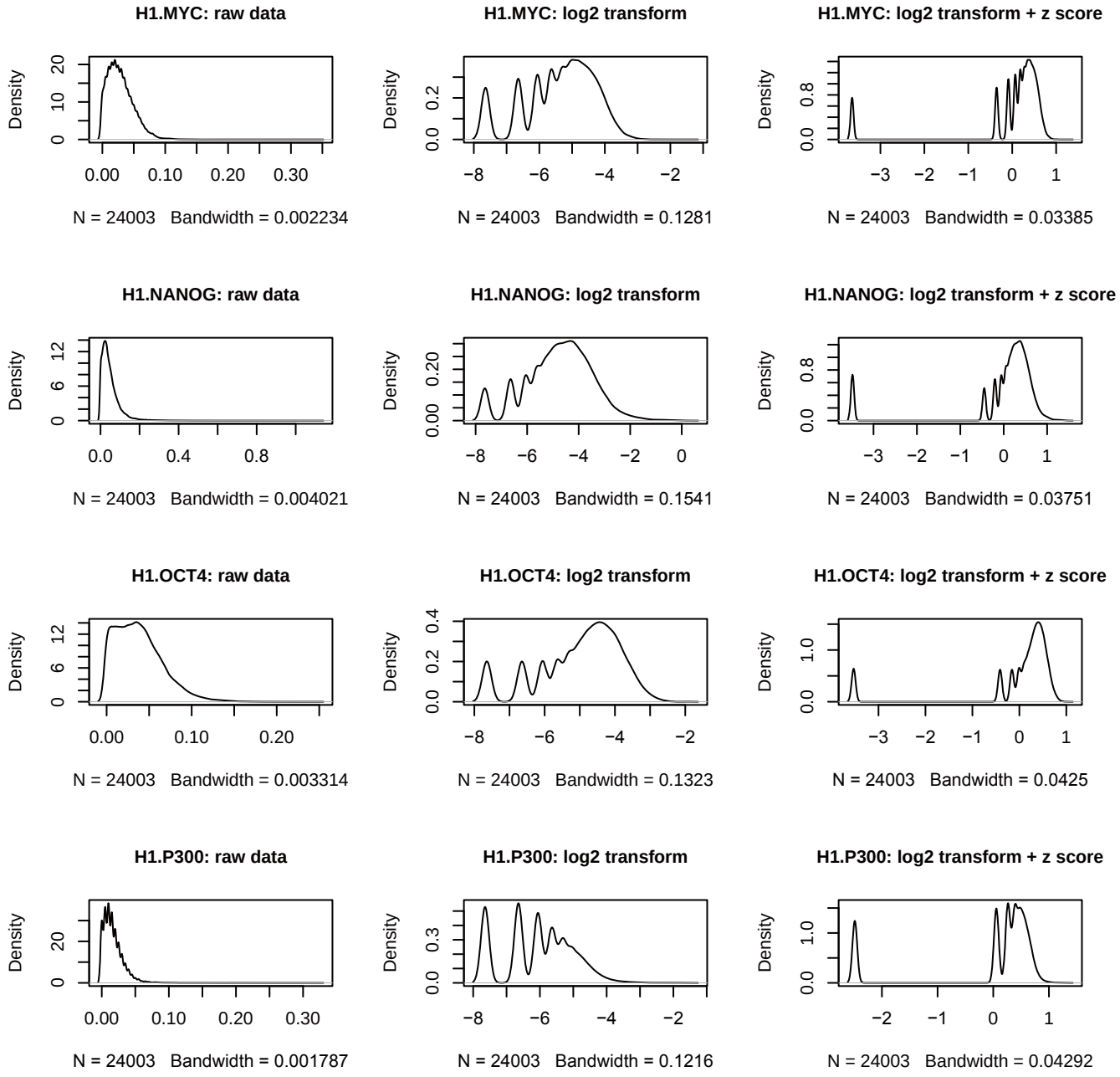

H1.MYC: raw data
Density
N = 24003 Bandwidth = 0.002234
H1.MYC: log2 transform
Density
N = 24003 Bandwidth = 0.1281
H1.MYC: log2 transform + z score
Density
N = 24003 Bandwidth = 0.03385
H1.NANOG: raw data
Density
N = 24003 Bandwidth = 0.004021
H1.NANOG: log2 transform
Density
N = 24003 Bandwidth = 0.1541
H1.NANOG: log2 transform + z score
Density
N = 24003 Bandwidth = 0.03751
H1.OCT4: raw data
Density
N = 24003 Bandwidth = 0.003314
H1.OCT4: log2 transform
Density
N = 24003 Bandwidth = 0.1323
H1.OCT4: log2 transform + z score
Density
N = 24003 Bandwidth = 0.0425
H1.P300: raw data
Density
N = 24003 Bandwidth = 0.001787
H1.P300: log2 transform
Density
N = 24003 Bandwidth = 0.1216
H1.P300: log2 transform + z score
Density
N = 24003 Bandwidth = 0.04292

Figure S3

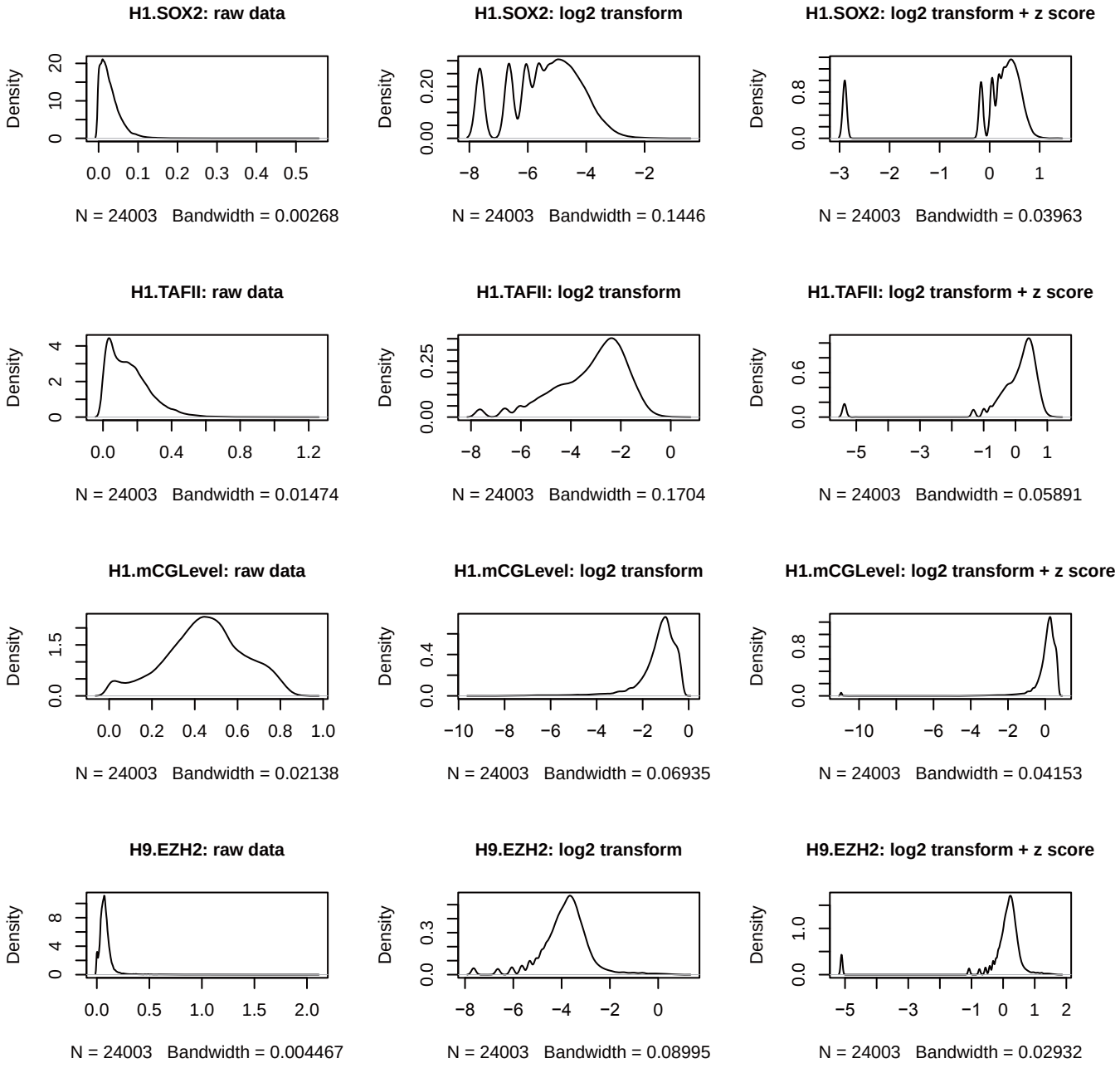

H1.SOX2: raw data
Density
N = 24003 Bandwidth = 0.00268
H1.SOX2: log2 transform
Density
N = 24003 Bandwidth = 0.1446
H1.SOX2: log2 transform + z score
Density
N = 24003 Bandwidth = 0.03963
H1.TAFII: raw data
Density
N = 24003 Bandwidth = 0.01474
H1.TAFII: log2 transform
Density
N = 24003 Bandwidth = 0.1704
H1.TAFII: log2 transform + z score
Density
N = 24003 Bandwidth = 0.05891
H1.mCGLevel: raw data
Density
N = 24003 Bandwidth = 0.02138
H1.mCGLevel: log2 transform
Density
N = 24003 Bandwidth = 0.06935
H1.mCGLevel: log2 transform + z score
Density
N = 24003 Bandwidth = 0.04153
H9.EZH2: raw data
Density
N = 24003 Bandwidth = 0.004467
H9.EZH2: log2 transform
Density
N = 24003 Bandwidth = 0.08995
H9.EZH2: log2 transform + z score
Density
N = 24003 Bandwidth = 0.02932

Figure S3

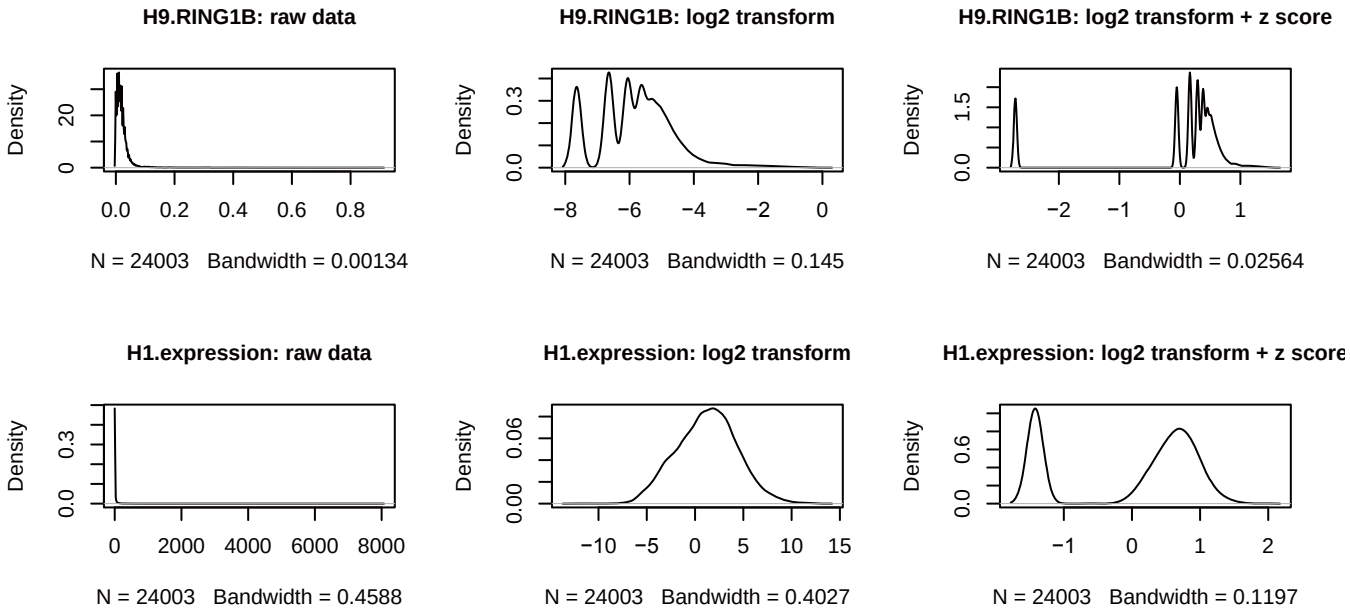


Figure S3. The effect of logarithmic and z-score transformation on sequence tag density within +/- 2kb of TSS. Left Column: the distribution of raw tag densities for each TF/modification; Middle Column: Tag density distributions after the logarithmic transformation; Right Column: Tag density distributions normalized by both the logarithmic and the z-score transformation.

Figure S4

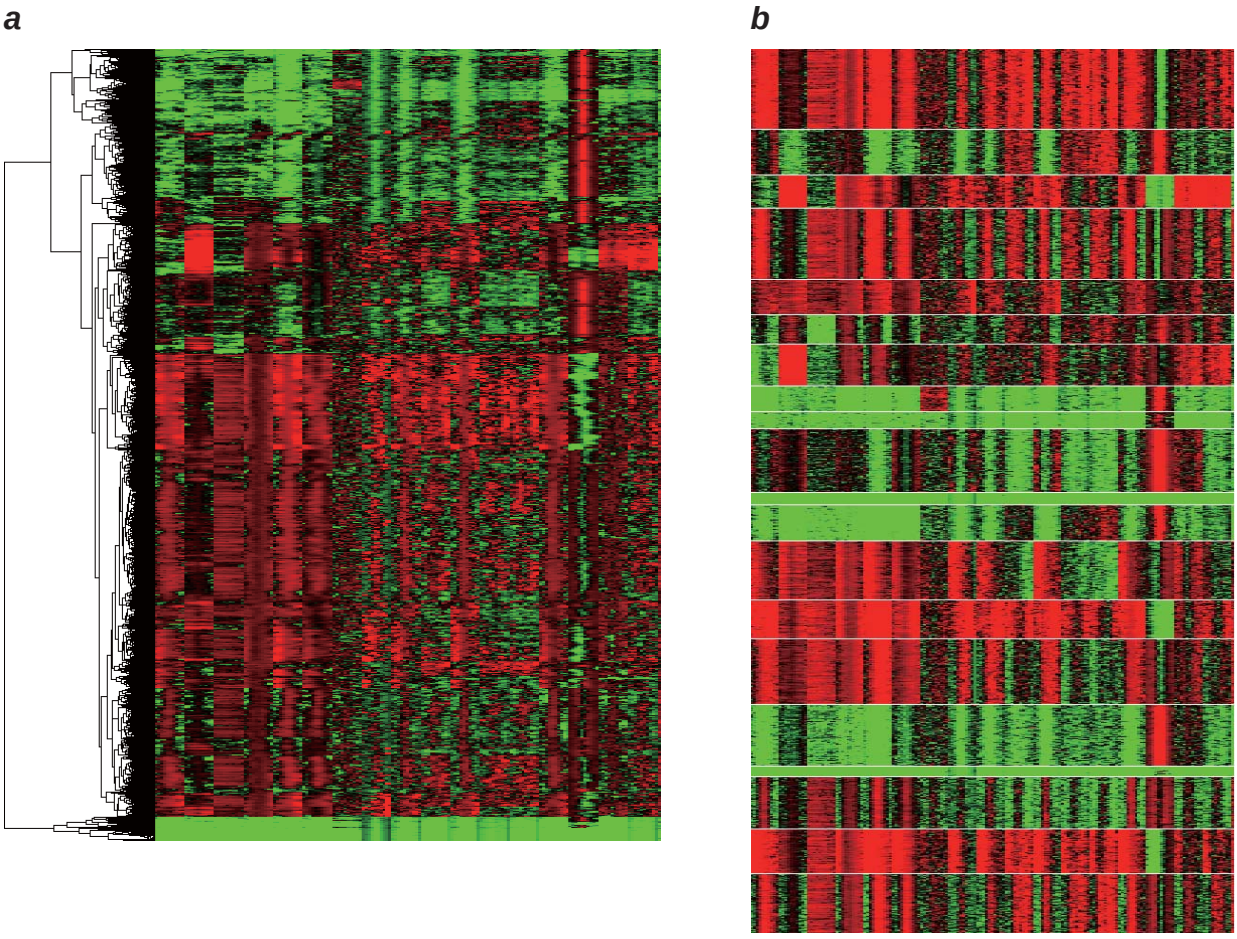
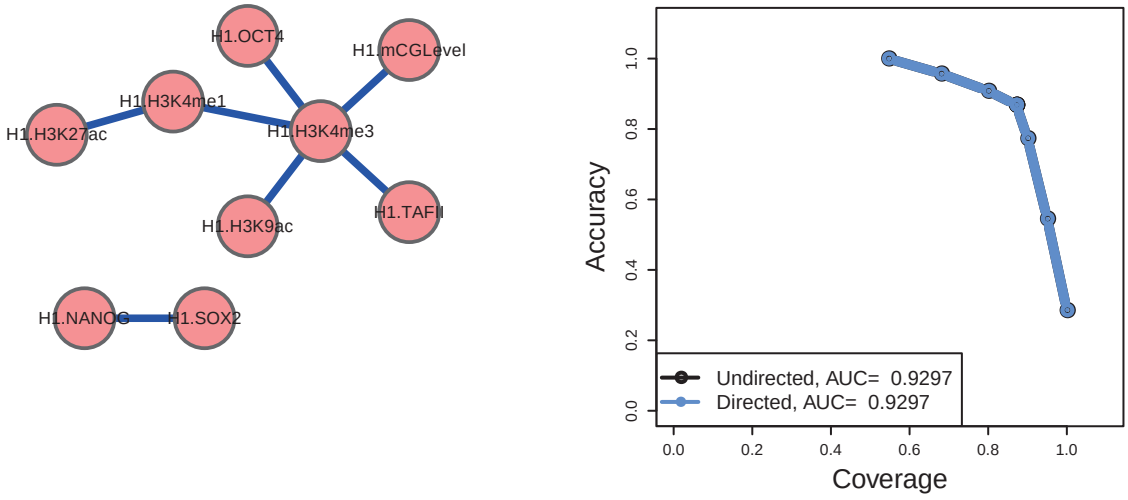


Figure S4. The effectiveness of profile-based clustering in reducing noise and extracting hidden biological 'states' from raw deep sequencing data. **(a)** Hierarchical clustering of the ChIP-Seq tag distributions around TSS. For each transcription factor / epigenetic modification, the tag distribution at [-2kb, +2kb] region around TSS is preprocessed and represented by a 10-bin vector used for BN learning (Methods), and the vectors for different factors are concatenated into a 171-bin long vector (including 1 bin for gene expression) before clustering. Red color represents larger values in the cells and green denotes smaller values. After clustering, many distinct clusters of genes emerge, which reflects well-defined biological states. **(b)** Same to (a), but the k-means algorithm in Cluster 3.0 is used instead of the hierarchical clustering algorithm, which is able to generate more compact clusters of data.

Figure S5

a



b

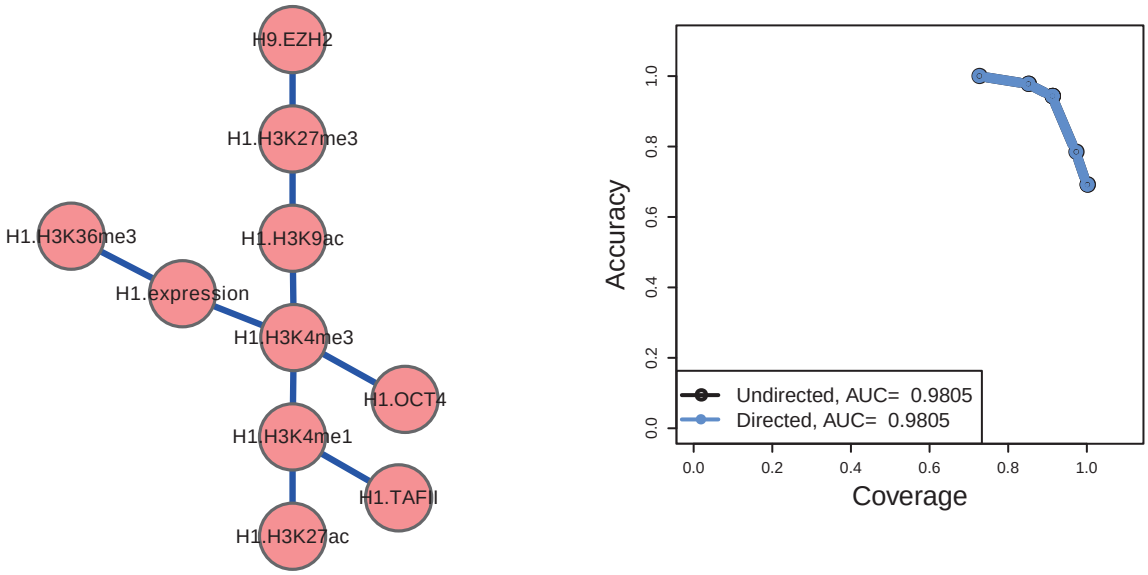
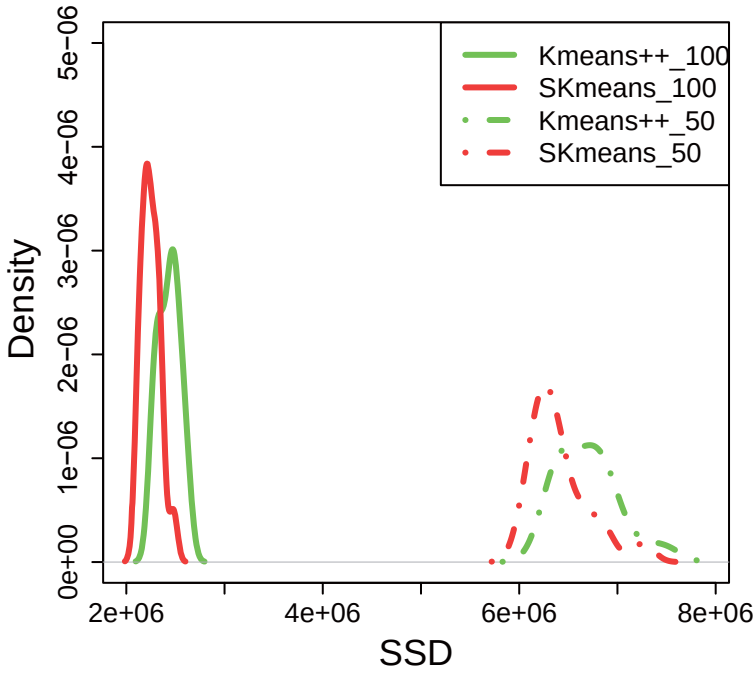


Figure S5. The consensus networks and the network stability curves when using alternative profile-based clustering algorithms to learn the hESC regulator network. Similar to Figure 1A, the proposed L1-RPS kernel and the Gaussian kernel are used to handle vectored and real-valued BN nodes, respectively. **(a)** The profiled cluster centers are generated by the ordinary k-means algorithm in Cluster 3.0, where the number of clusters is 1000 for the 10 datasets in the data re-sampling procedure for deriving the consensus network. **(b)** Similar to (a), except the profiled cluster centers are generated by the affinity propagation algorithm. In this algorithm, it is not possible to set the number of clusters explicitly ahead of execution. However, we have fine-tuned the 'preference' parameter so that the number of clusters it generates for each dataset is very close to 1000.

Figure S6

a


Kmeans++_100
SKmeans_100
Kmeans++_50
SKmeans_50
Density
0e+00
1e−06
2e−06
3e−06
4e−06
5e−06
2e+06
4e+06
6e+06
8e+06
SSD

b

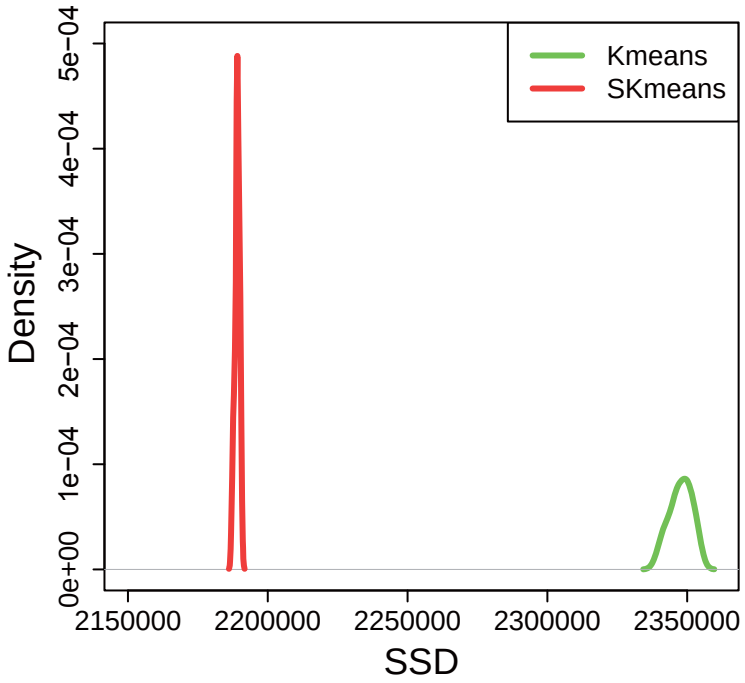
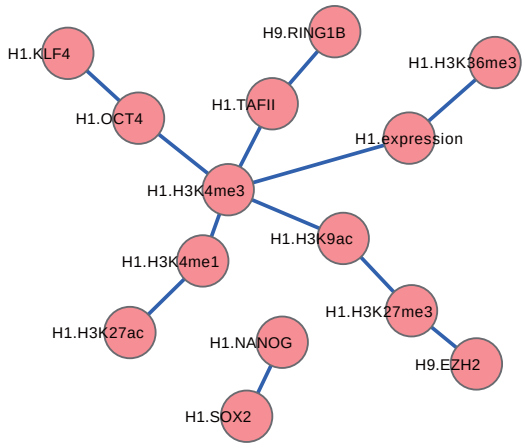
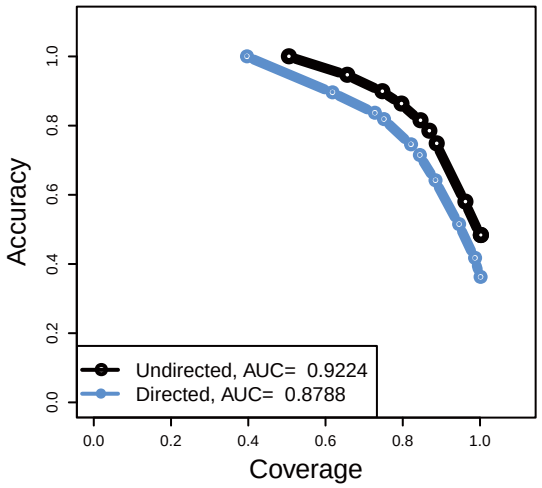

Kmeans
SKmeans
Density
0e+00
1e−04
2e−04
3e−04
4e−04
5e−04
2150000
2200000
2250000
2300000
2350000
SSD

Figure S6. Comparing the Super k-means algorithm with the k-means++ and the ordinary k-means algorithm. **(a)** Comparing the Super k-means with the k-means++ algorithm on clustering the "spam_input.txt" data (contained in the k-means++ software) into 50 and 100 groups. Distributions of the sum of squared Euclidean distances (SSD) from data to cluster center in 20 runs are plotted for each algorithm. Red/Green color denote the "Super k-means" / "k-means++" algorithm, and solid/dashed lines denote the two tasks of clustering data into 100/50 clusters, respectively. **(b)** Comparing the Super k-means algorithm with the ordinary k-means algorithm in Cluster 3.0 for grouping the concatenated gene-wise tag profiles into 1000 clusters. This is exactly the profile-based clustering task for learning the hESC regulator network in Figure 1A. Again, both of the two algorithms were run 20 times and the Euclidean distance was used for clustering. In the resulting SSD distributions, Red/Green color denotes the "Super k-means" / "ordinary k-means" algorithm, respectively.

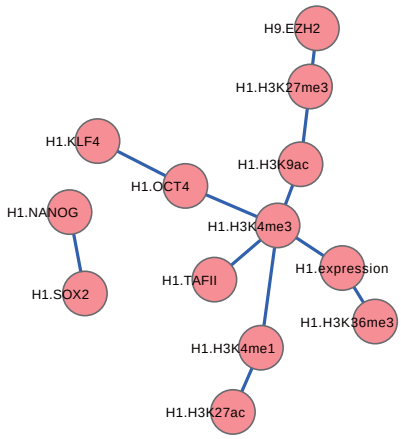
Figure S7 a

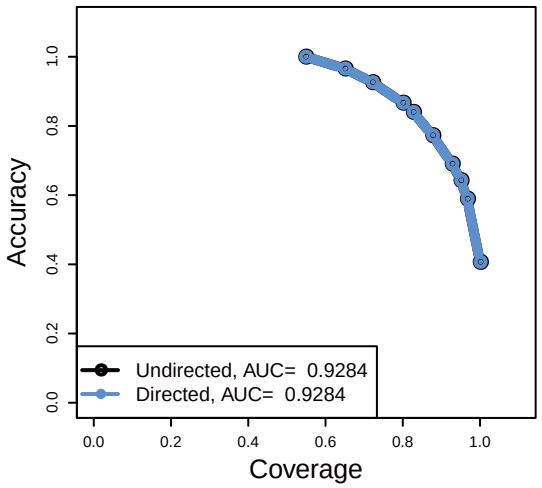
lambda = 2



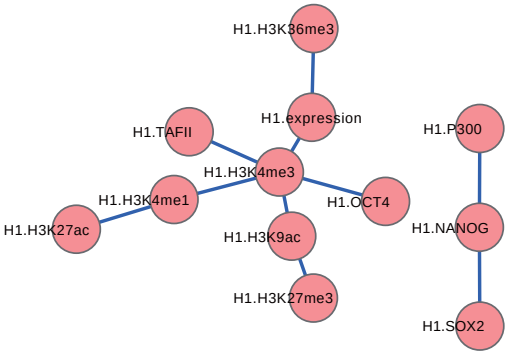


lambda = 3





lambda = 4



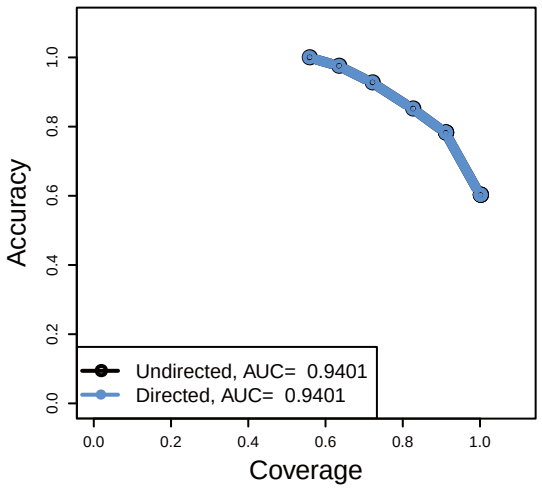


Figure S7 b

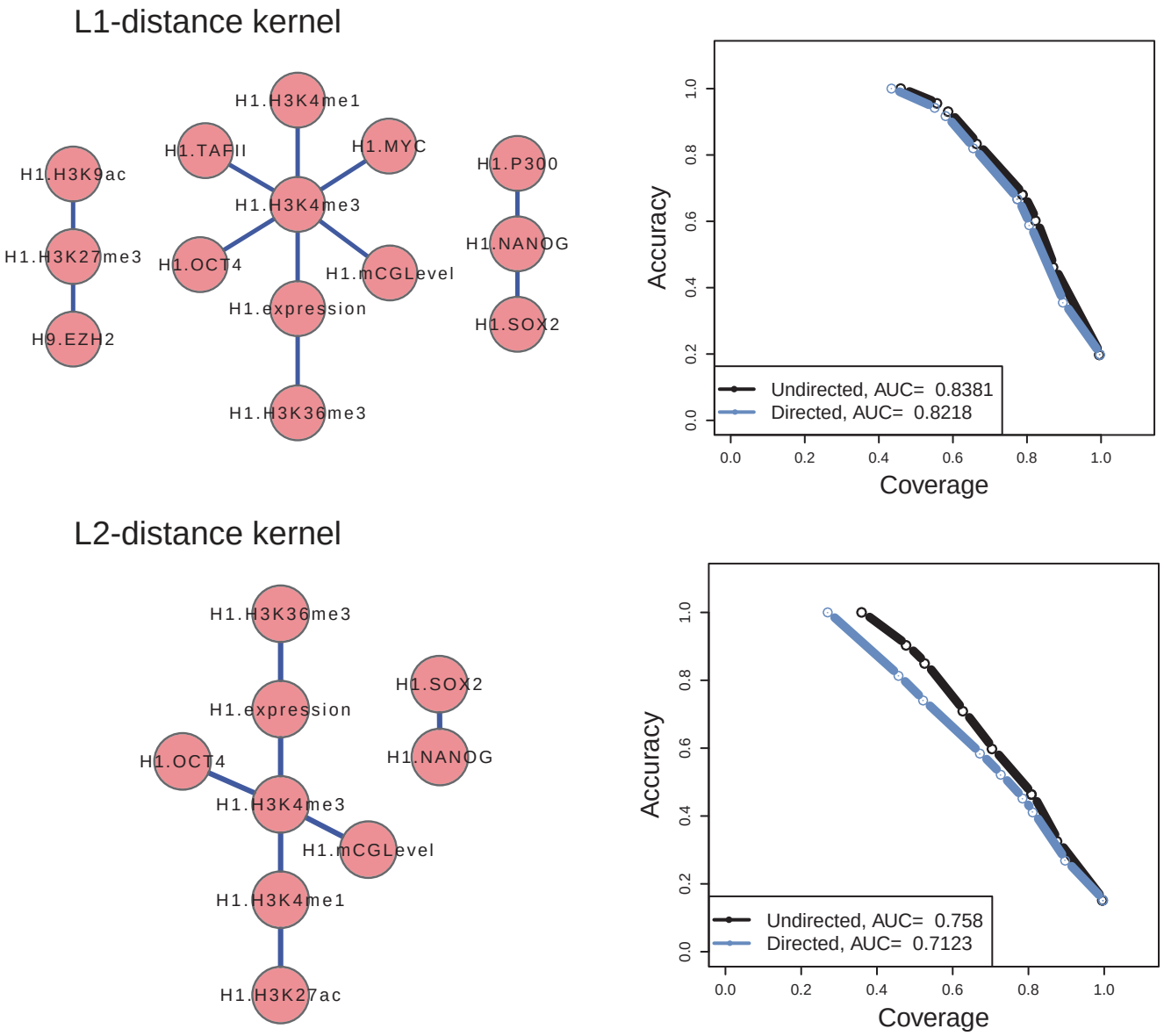


Figure S7. The performance of using alternative kernels in SeqSpider on profiled data **(a)** Using the time-warping distance kernel for vectored data in Bayesian network learning. In the BN inference procedure, the time-warping distance [31] replaced the L1-RPS distance to define the kernel for vectored data and the other settings are kept the same as learning the hESC regulator network in Figure 1A. By performing 10-fold data re-sampling based network learning procedure with three different weights of the complexity term in the BIC scoring criterion ($\lambda = 2,3,4$), the consensus network and the network stability curves for $\lambda = 2$; $\lambda = 3$; $\lambda = 4$ are obtained. **(b)** Stability curves and consensus networks obtained by using the standard L1/L2 distance to define the kernel for vectored data. Exactly the same parameter setting for learning the hESC network in Figure 1A is used here ($\lambda = 3$).

Figure S8

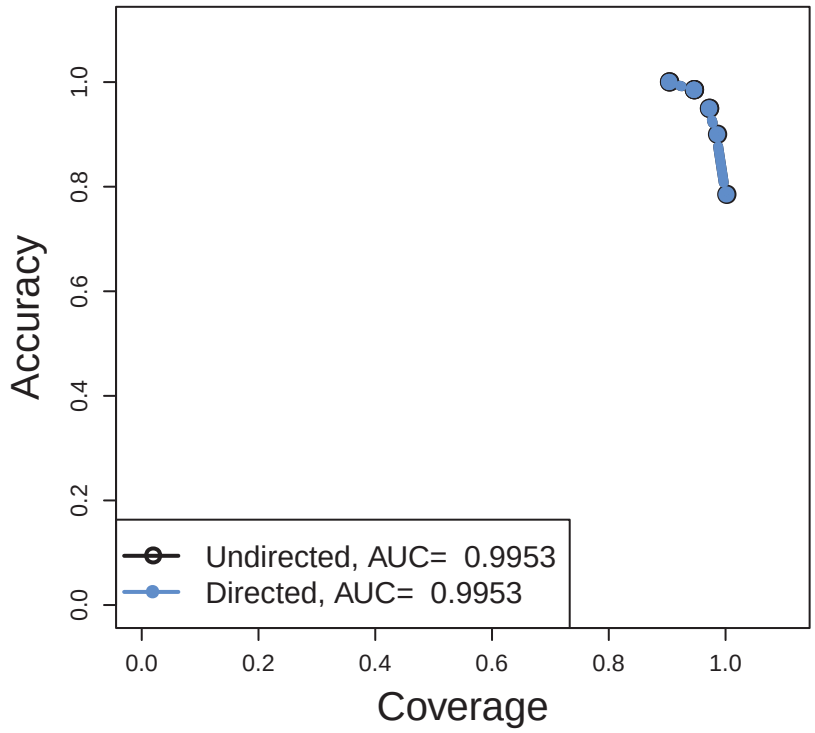
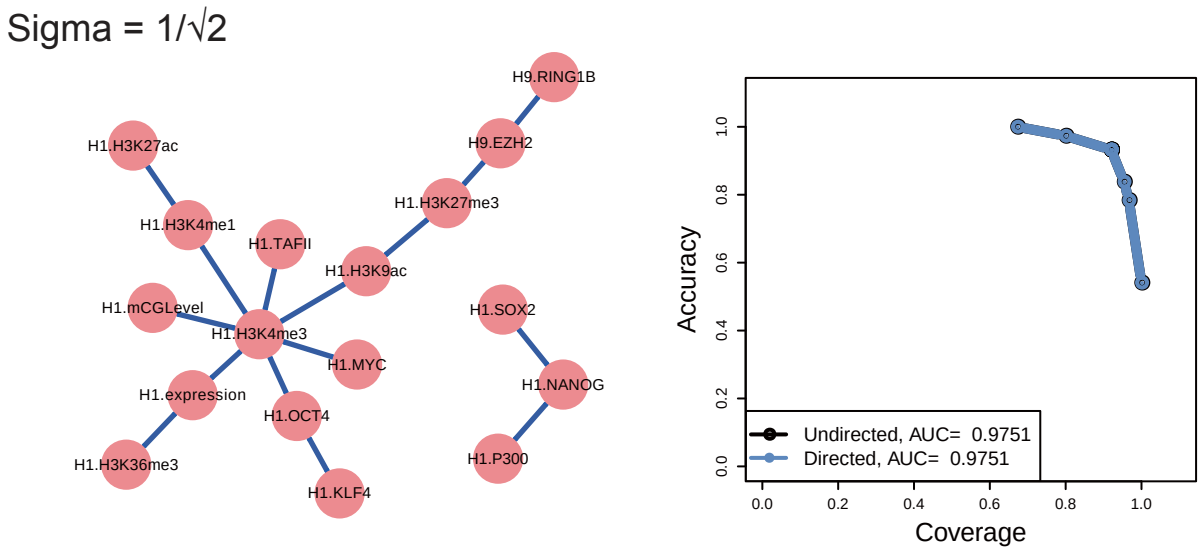


Figure S8. Network stability curves for comparing the consensus hESC regulator networks derived from the data re-sampling based network learning procedure under different regularization strengths (The weight parameter λ in the BN scoring function was set to {2.0, 2.25, 2.5…4.0}). Except this difference, all the other experimental settings are the same as learning the hESC regulator network in Figure 1A.

Figure S9

a



b

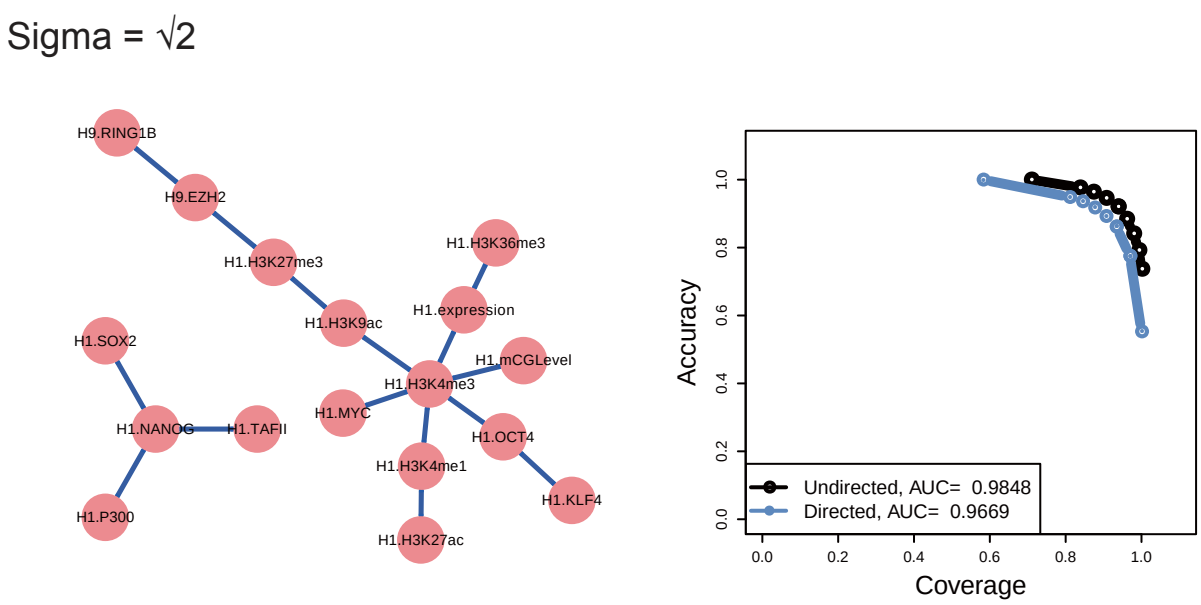
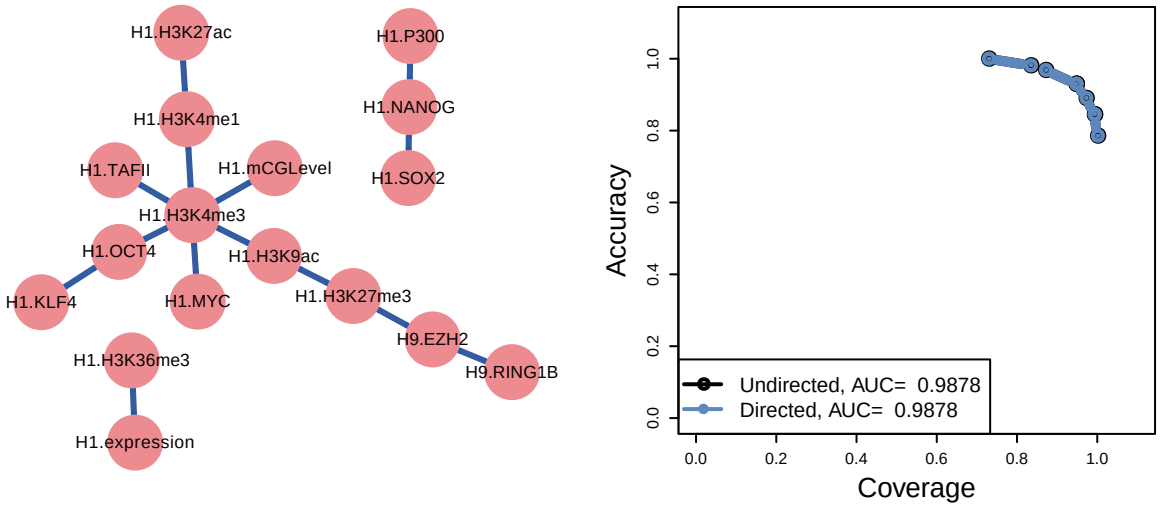


Figure S9. The impact of kernel width on Bayesian network structure inference. Fixing all the other experimental settings the same as learning the hESC regulator network in Figure 1A, two data re-sampling based network learning procedure are performed with decreased / increased kernel widths, where the default kernel widths (specified by the 'wise normalization' approach) for all nodes in the BN are scaled by **(a)** $1/\sqrt{2}$ and **(b)** $\sqrt{2}$, respectively. Shown in the two panels are the consensus networks and the network stability curves of the two experiments.

Figure S10

a

CV percentage = 70%



b

CV percentage = 80%

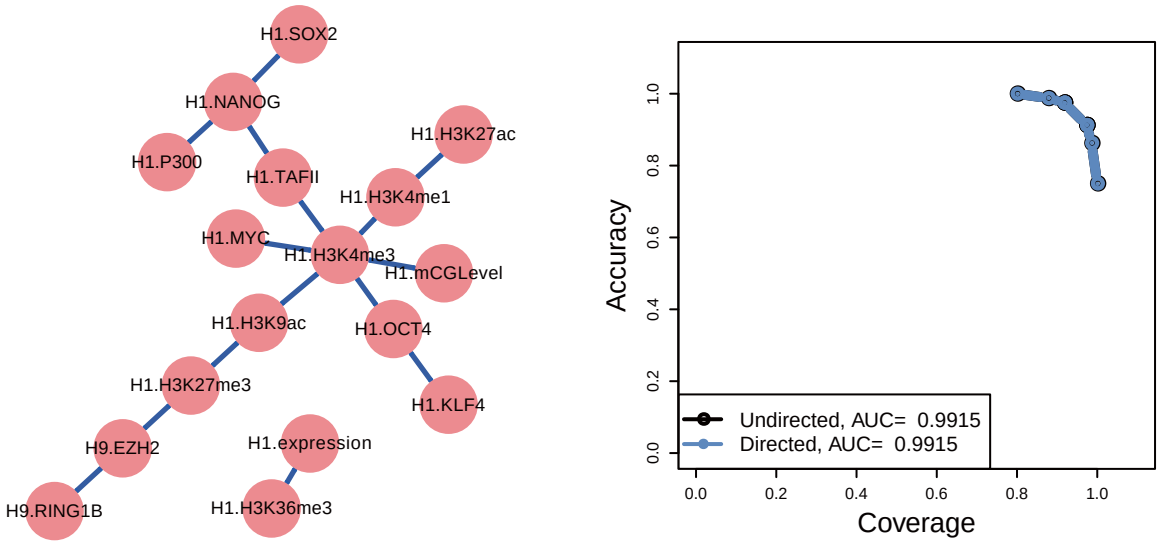
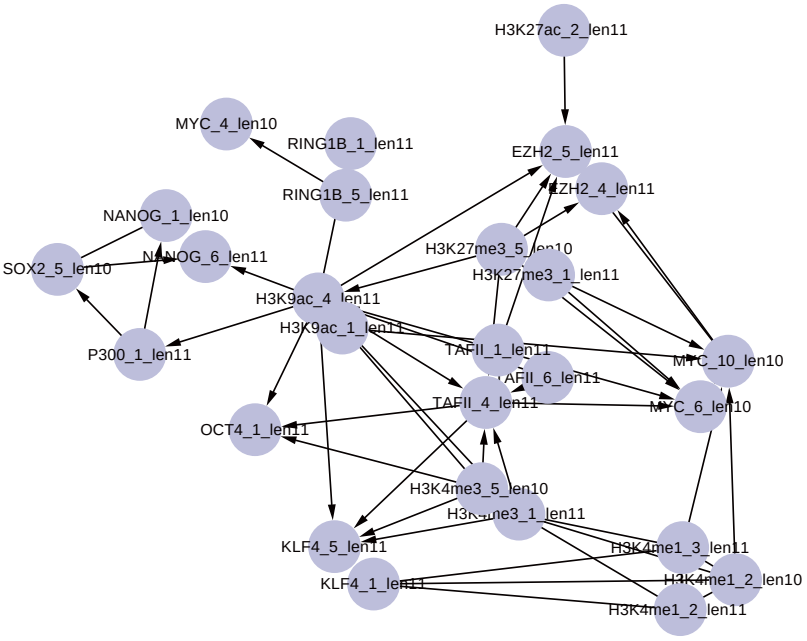


Figure S10. Stability of the SeqSpider algorithm for network inference. The 10-fold data re-sampling procedure for inferring the hESC regulator network in Figure 1A is repeated here, but fewer training data is used for each fold. **(a)** The consensus network and the network stability curve when each gene-wise training data has an independent 70% probability to be included in each fold; **(b)** the same to (a), but with 80% probability.

Figure S11

a



b

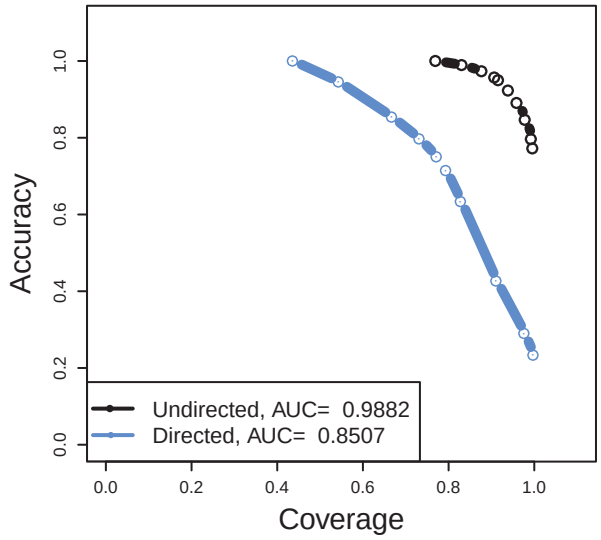


Figure S11. Inference of a general motif-motif interaction network, where the hESC regulator network (Figure 1A) is not used as a prior to constrain the structure search. **(a)** The consensus motif-motif interaction network derived from the 10-fold data re-sampling based network learning procedure, where edges appear ≥ 8 times in the 10 PDAGs are included (to arrive at approximately the same number of edges as the hESC context-specific motif network in Figure S22a). Each motif name is composed of three components, the first is the name of the corresponding hESC regulator, the second is the rank of the motif in the DME2 output and the last is the length of the motif itself. **(b)** The network stability curves based on directed / undirected edges.

Figure S12

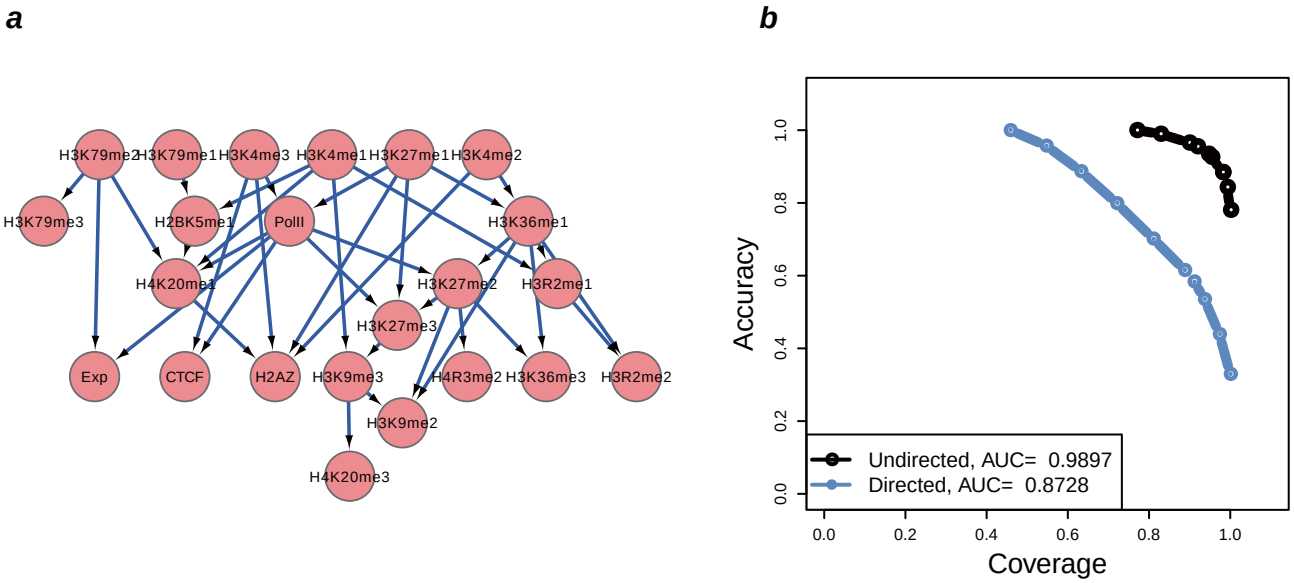
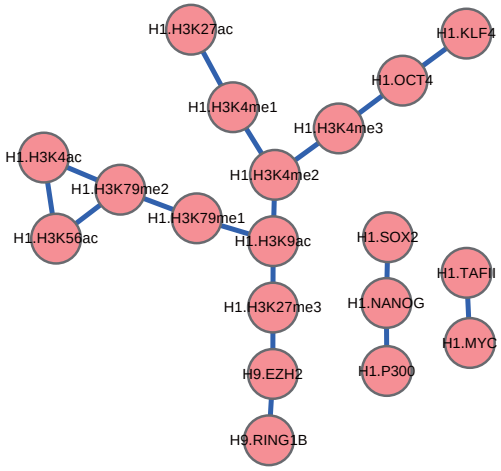
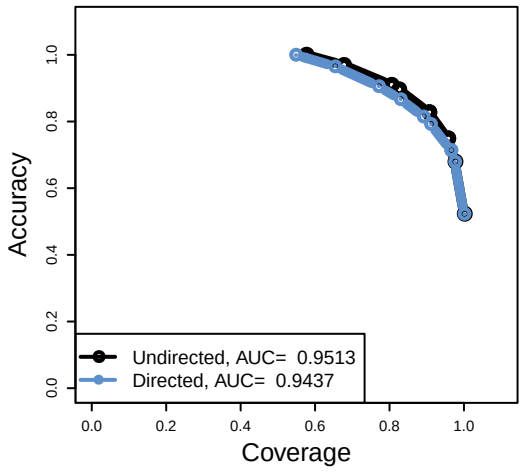


Figure S12. Inferring regulatory network from the ChIP-Seq data of CD4+ T cells using SeqSpider. Different from a previous study [8] where discretized total tag counts in the TSS +/- 1kb region was used, SeqSpider leverages on the more informative tag profiles in the same genomic region for BN inference. **(a)** The consensus regulatory network inferred by SeqSpider through the data re-sampling based network learning procedure, where edges appear ≥ 7 times in the 10 PDAGs were included. **(b)** The network stability curves based on directed / undirected edges.

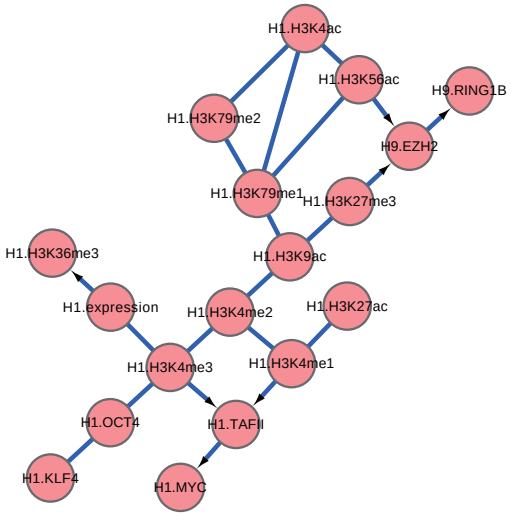
Figure S13 a

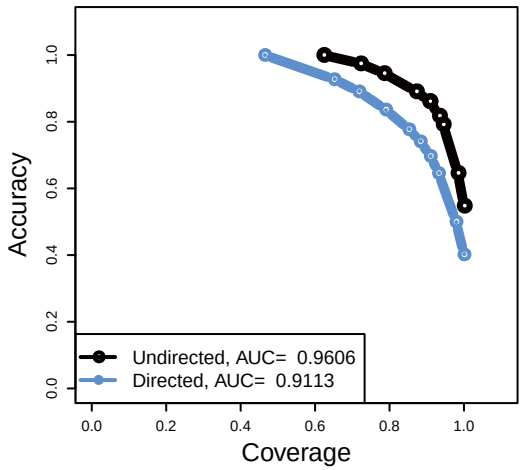
k= 1000, repeat =20



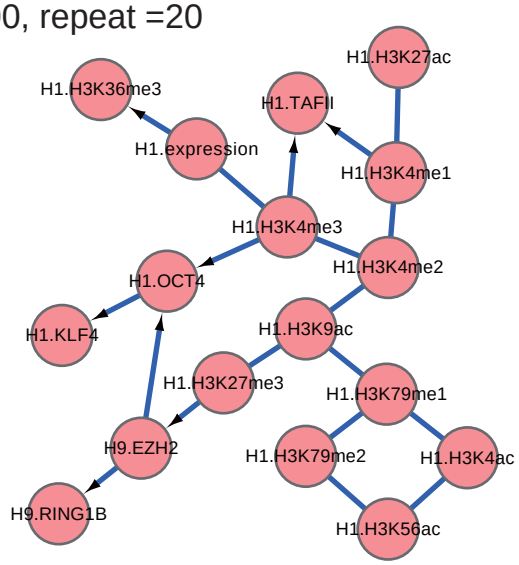


k= 1500, repeat =20





k= 2000, repeat =20



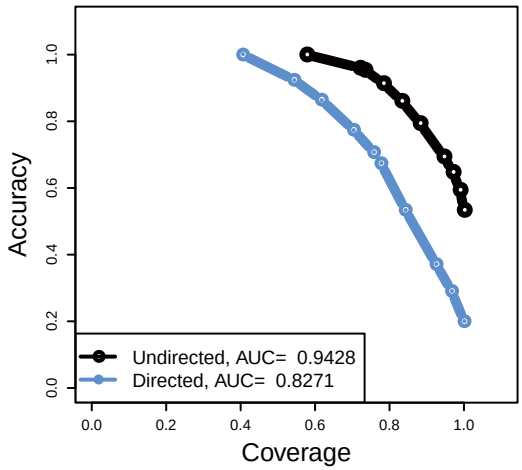


Figure S13 b

The 'reduced' network

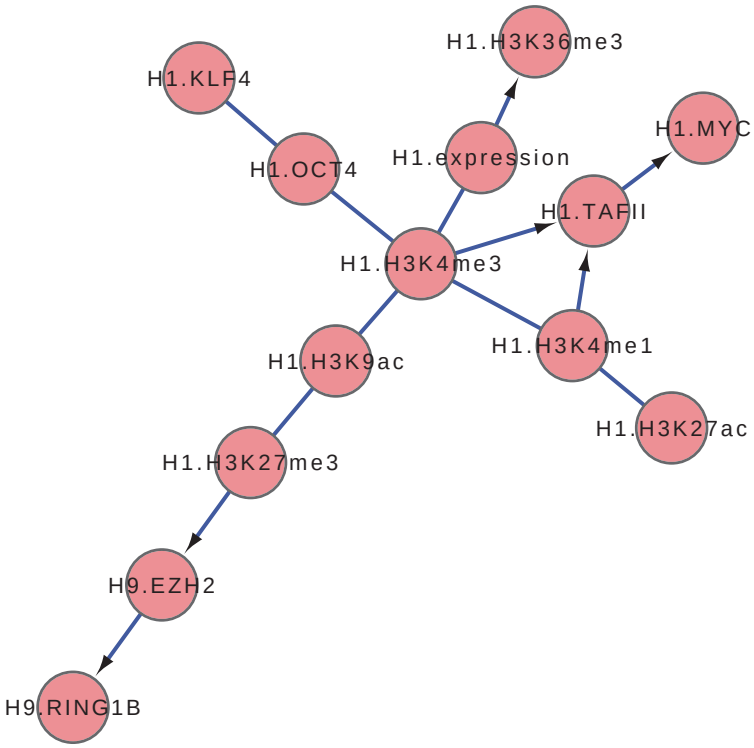


Figure S13. The hESC regulator network learned from an extended set of deep sequencing data. Five more histone marks (H3K4ac, H3K4me2, H3K56ac, H3K79me1 and H3K79me2) are included to infer the hESC regulator network. **(a)** The consensus network and the network stability curves (directed / undirected edge) of the BNs learned by SeqSpider on 1000 profile-based cluster centers (k=1000); on 1500 cluster centers (k=1500); on 2000 cluster centers (k=2000). All the other experimental settings are the same as learning the hESC regulator network in Figure 1A. **(b)** The 'reduced network' for the updated network derived from 1500 cluster centers (k=1500), where we have removed the 5 newly added nodes in order to systematically compare it with the hESC network in Figure 1A.

Figure S14

a

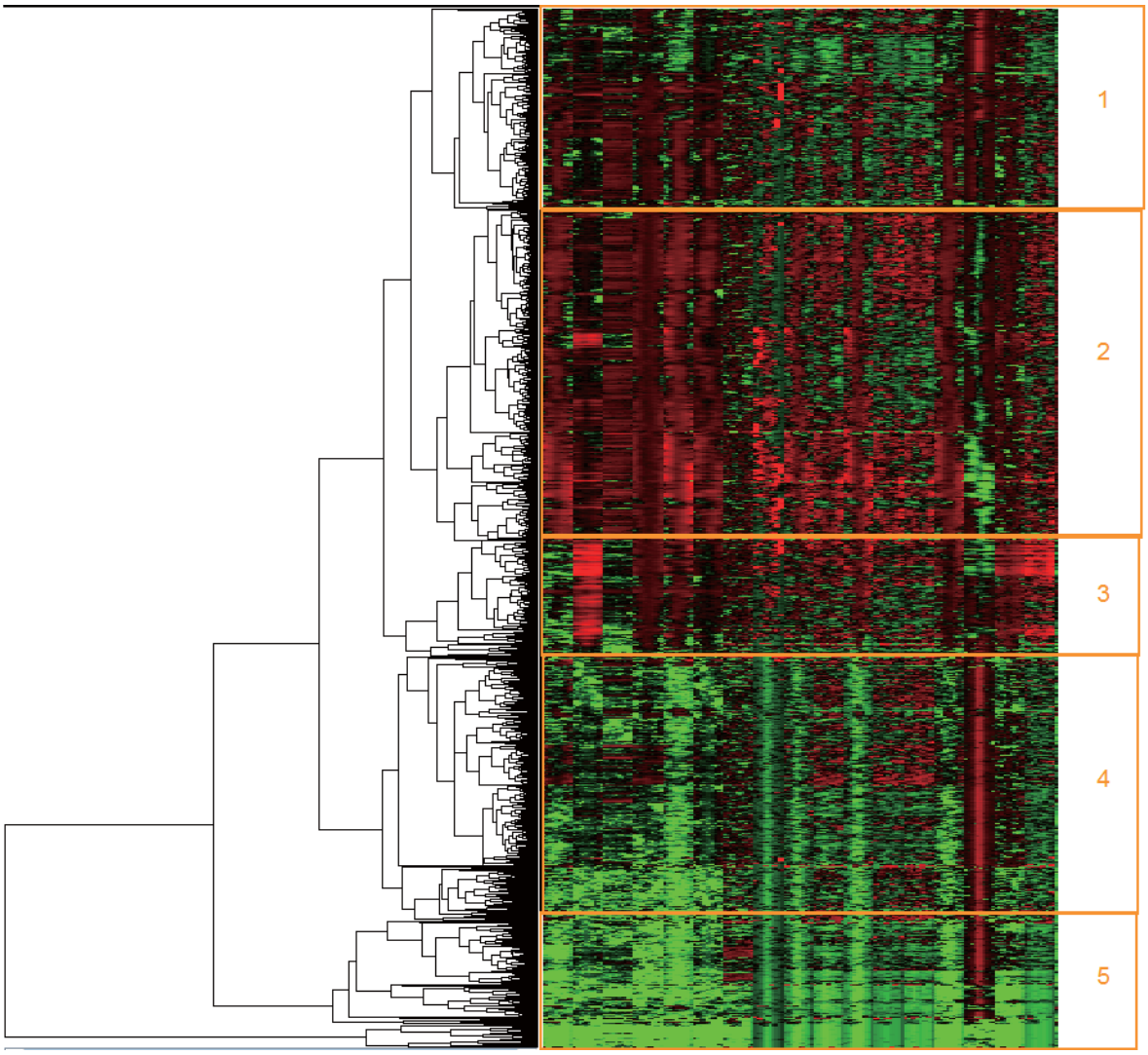
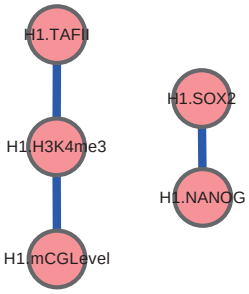
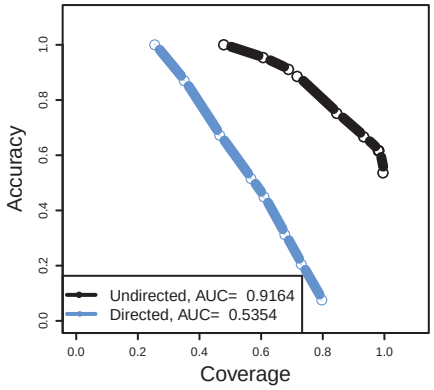


Figure S14

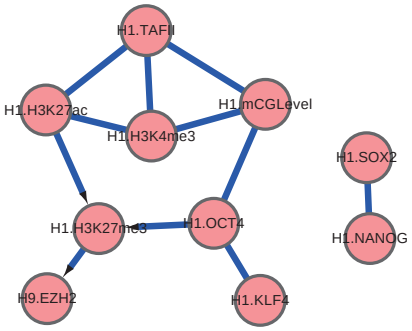
b

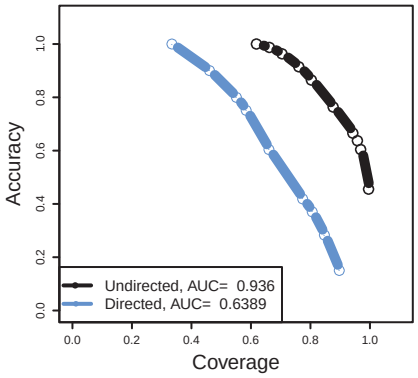
Cluster 1



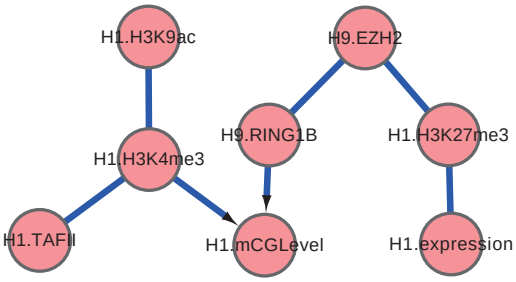


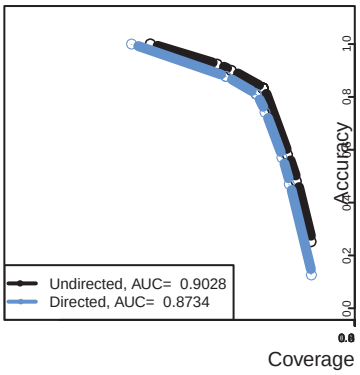
Cluster 2



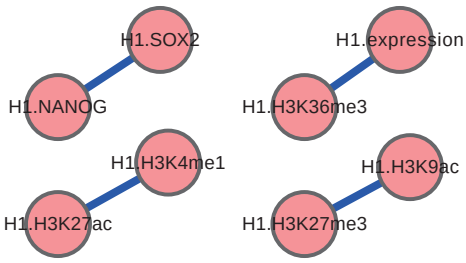


Cluster 3





Cluster 4



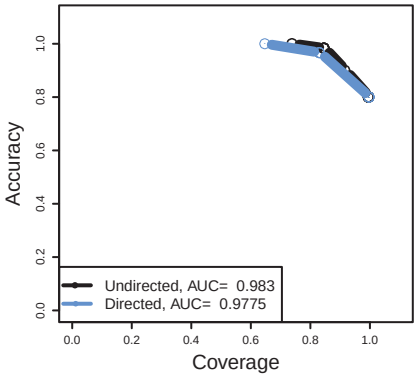
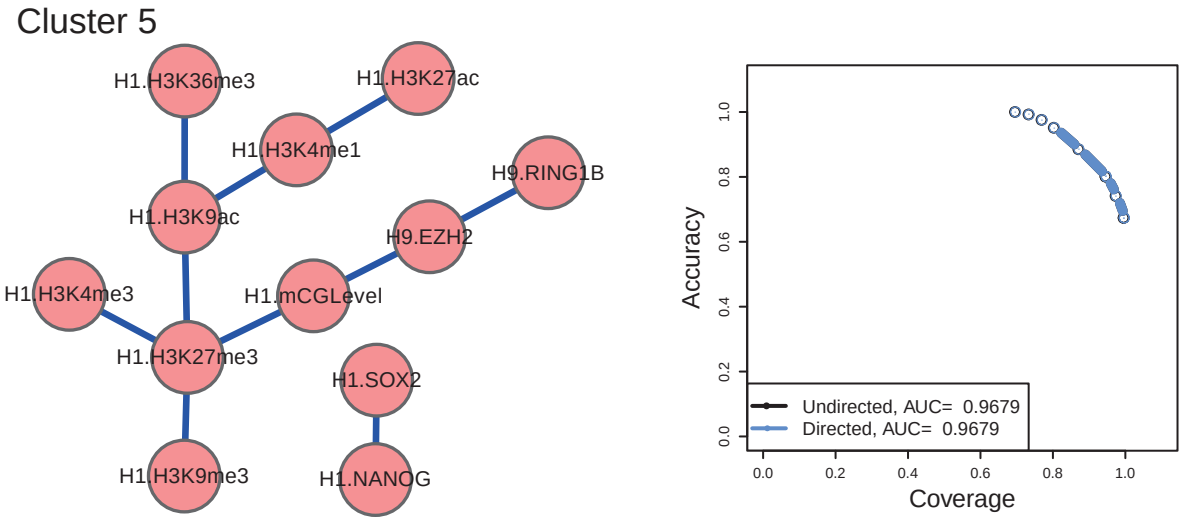


Figure S14

b



c

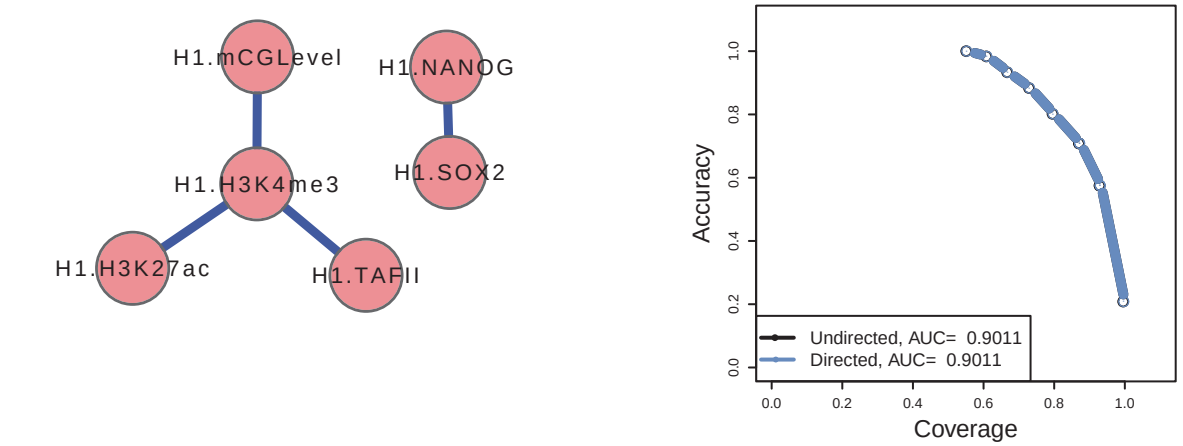


Figure S14. Regulatory relationships within different groups of promoters distinguished by deep sequencing tag profiles in hESC. The 1000 cluster centers (from profile-based clustering) for learning the hESC regulator network in Figure 1A is further divided into large groups where the TF/epigenetic patterns in each group are similar. **(a)** Hierarchical clustering of the 1000 representative tag profiles using Cluster 3.0. Five distinct groups of promoters are marked. **(b)** A regulatory network is inferred for each group of promoters to uncover the local regulations/interactions therein. Gene-wise tag profiles that map to the cluster centers were used as training data to infer the local regulatory network in that group. The consensus networks and the network stability curves for groups 1 to 5 are shown next to the respective networks. **(c)** Re-learning a regulatory network for group 2 based on 1000 Super k-means cluster centers derived from concatenated gene-wise profiles in that group. The consensus network and the network stability curves are presented.

Figure S15

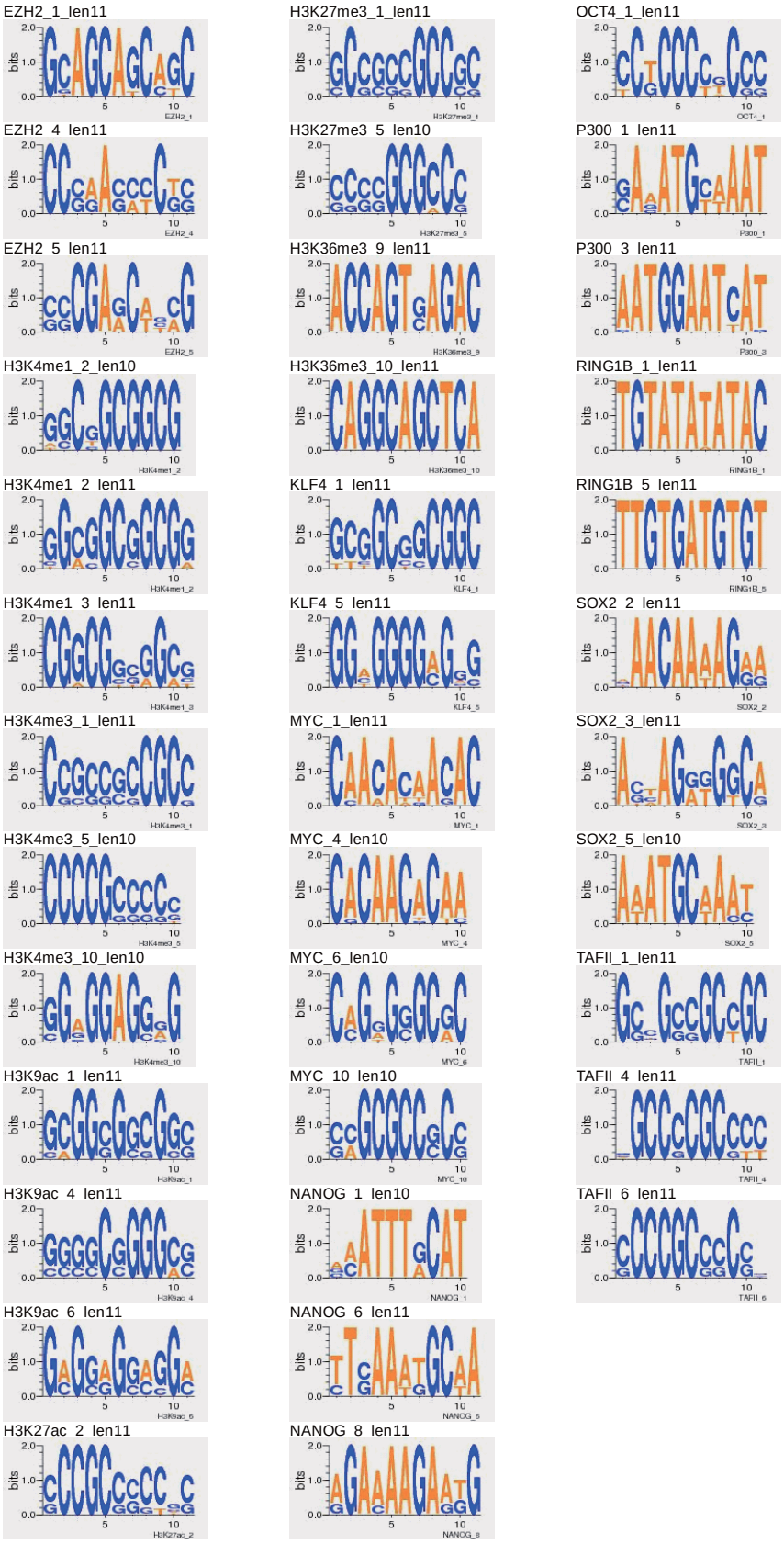

EZH2_1_len11
H3K27me3_1_len11
OCT4_1_len11
EZH2_4_len11
H3K27me3_5_len10
P300_1_len11
EZH2_5_len11
H3K36me3_9_len11
P300_3_len11
H3K4me1_2_len10
H3K36me3_10_len11
RING1B_1_len11
H3K4me1_2_len11
KLF4_1_len11
RING1B_5_len11
H3K4me1_3_len11
KLF4_5_len11
SOX2_2_len11
H3K4me3_1_len11
MYC_1_len11
SOX2_3_len11
H3K4me3_5_len10
MYC_4_len10
SOX2_5_len10
H3K4me3_10_len10
MYC_6_len10
TAFII_1_len11
H3K9ac_1_len11
MYC_10_len10
TAFII_4_len11
H3K9ac_4_len11
NANOG_1_len10
TAFII_6_len11
H3K9ac_6_len11
NANOG_6_len11
H3K27ac_2_len11
NANOG_8_len11

Figure S15. Distinctive sequence motifs enriched in the peak regions of ChIP-Seq signals. The position weight matrices of these motifs are visualized using Web-Logo. Each motif name is composed of three components, the first is the name of the corresponding hESC regulator, the second is the rank of the motif in the DME2 output and the last is the length of the motif itself. To avoid redundancy/overlaps, for each ChIP-seq experiment, only the 2~3 most representative motifs out of the top 10 DME2 output motifs are selected and illustrated here (which was then used to learn the motif-motif interaction network).

Figure S16

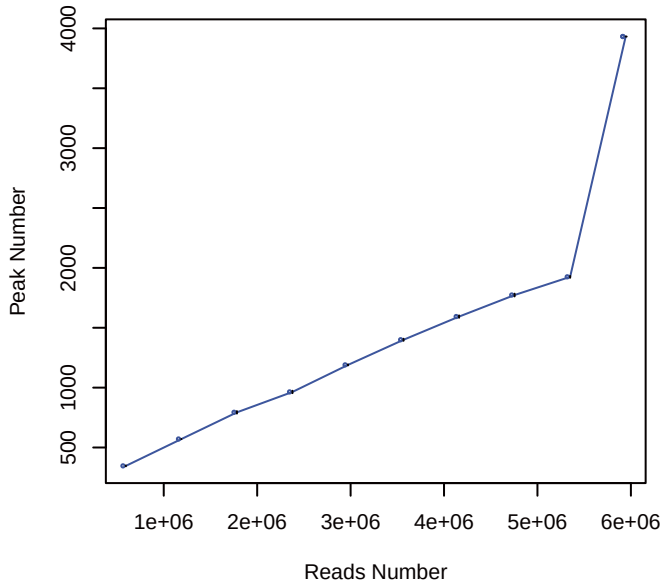


Figure S16. The number of H3K9me3 peaks detected by SICER for different numbers of sequence tags. The X-axis indicates the fraction of total short cDNA tags which were randomly selected from the H3K9me3 ChIP-Seq dataset. The Y-axis represents the number of peaks detected by the SICER software under P-value cutoff 1e-6. The number of peaks continues to sharply grow with the fraction of sequence tags sampled, indicating that the tags are clearly unsaturated and the sequencing depth is below a sufficient coverage of the genome.

Figure S17

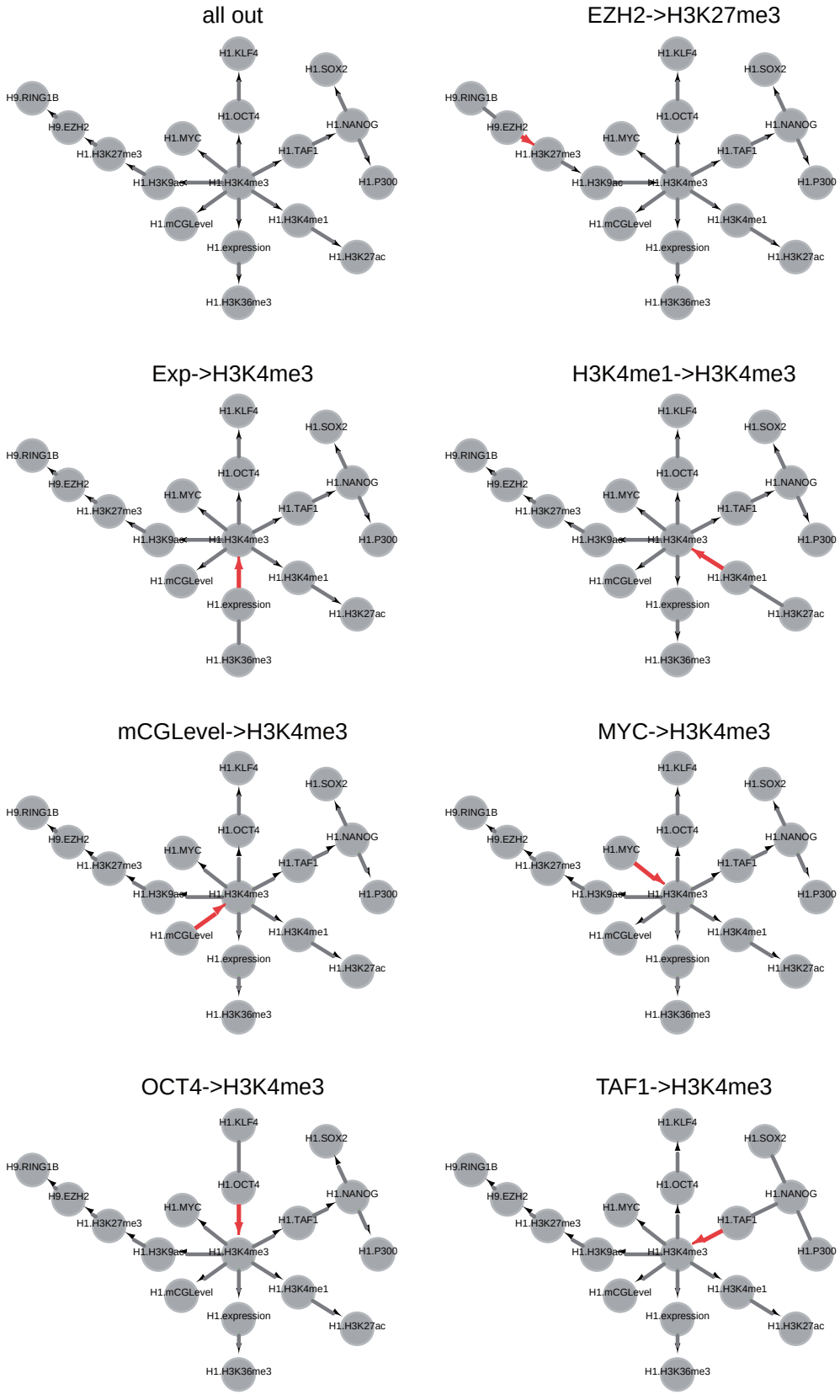

all out
EZH2->H3K27me3
Exp->H3K4me3
H3K4me1->H3K4me3
mCGLevel->H3K4me3
MYC->H3K4me3
OCT4->H3K4me3
TAF1->H3K4me3
H1.KLF4
H1.SOX2
H1.OCT4
H1.NANOG
H1.P300
H9.RING1B
H9.EZH2
H1.MYC
H1.TAF1
H1.H3K27me3
H1.H3K9ac
H1.H3K4me3
H1.mCGLevel
H1.H3K4me1
H1.expression
H1.H3K27ac
H1.H3K36me3

Figure S17. Global causal configurations of the hESC regulator network (Figure 1A). Based on the semantics of the BN structure, at most one edge connected to H3K4me3 is allowed to orient inwards (otherwise induce an immorality). Specifically, given the directionality of an edge in the inward branch to H3K4me3 (marked red), the influence will propagate to orient all edges in the other branches outward. See Note 5 for the biological implication about the global causal semantics.

Figure S18

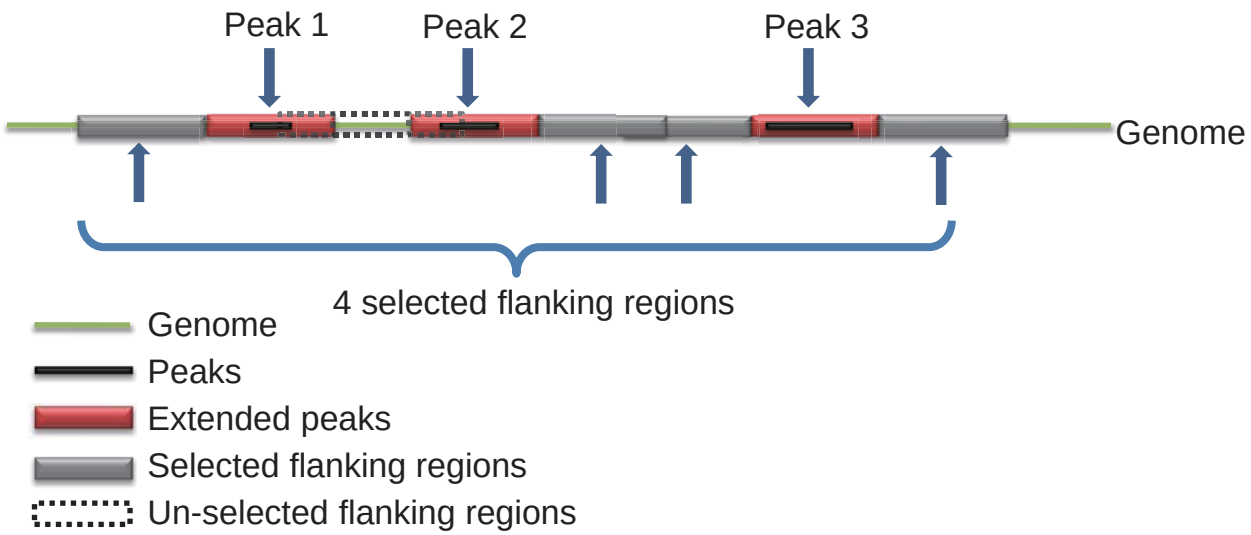


Figure S18. The choice of positive and negative sequences for motif discovery by DME2. The genome is represented by a green line and peaks of the tag distribution are marked by black segments (positive examples). Then, each peak is extended from its center to the 95% quantile length of all the peaks derived from this deep-sequencing experiment (red blocks). Finally, the gray blocks that flank and with the same length of each extended peak are defined to be negative examples given they do not overlap with any peak regions.

Figure S19

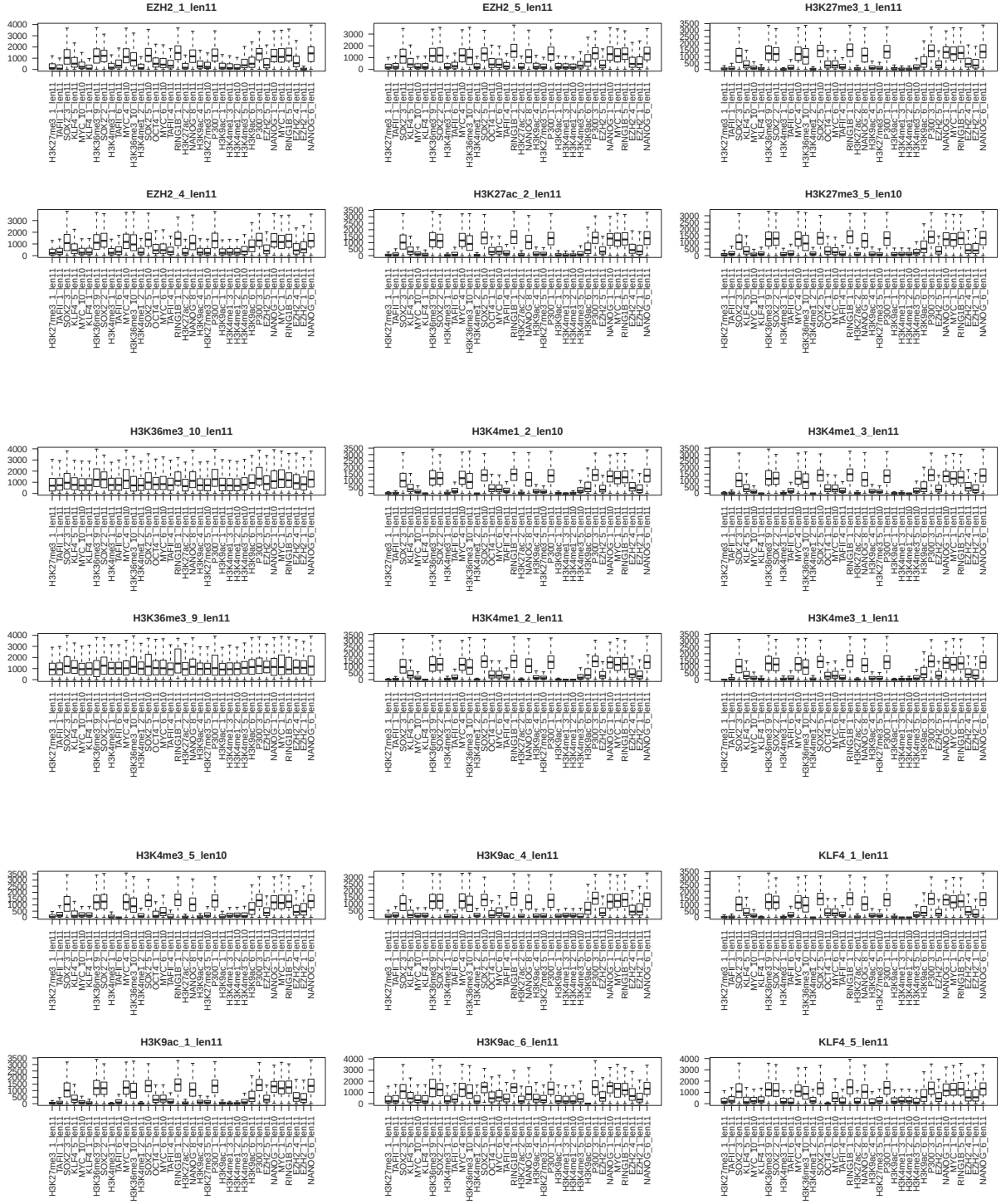

EZH2_1_len11
EZH2_5_len11
H3K27me3_1_len11
EZH2_4_len11
H3K27ac_2_len11
H3K27me3_5_len10
H3K36me3_10_len11
H3K4me1_2_len10
H3K4me1_3_len11
H3K36me3_9_len11
H3K4me1_2_len11
H3K4me3_1_len11
H3K4me3_5_len10
H3K9ac_4_len11
KLF4_1_len11
H3K9ac_1_len11
H3K9ac_6_len11
KLF4_5_len11

Figure S19

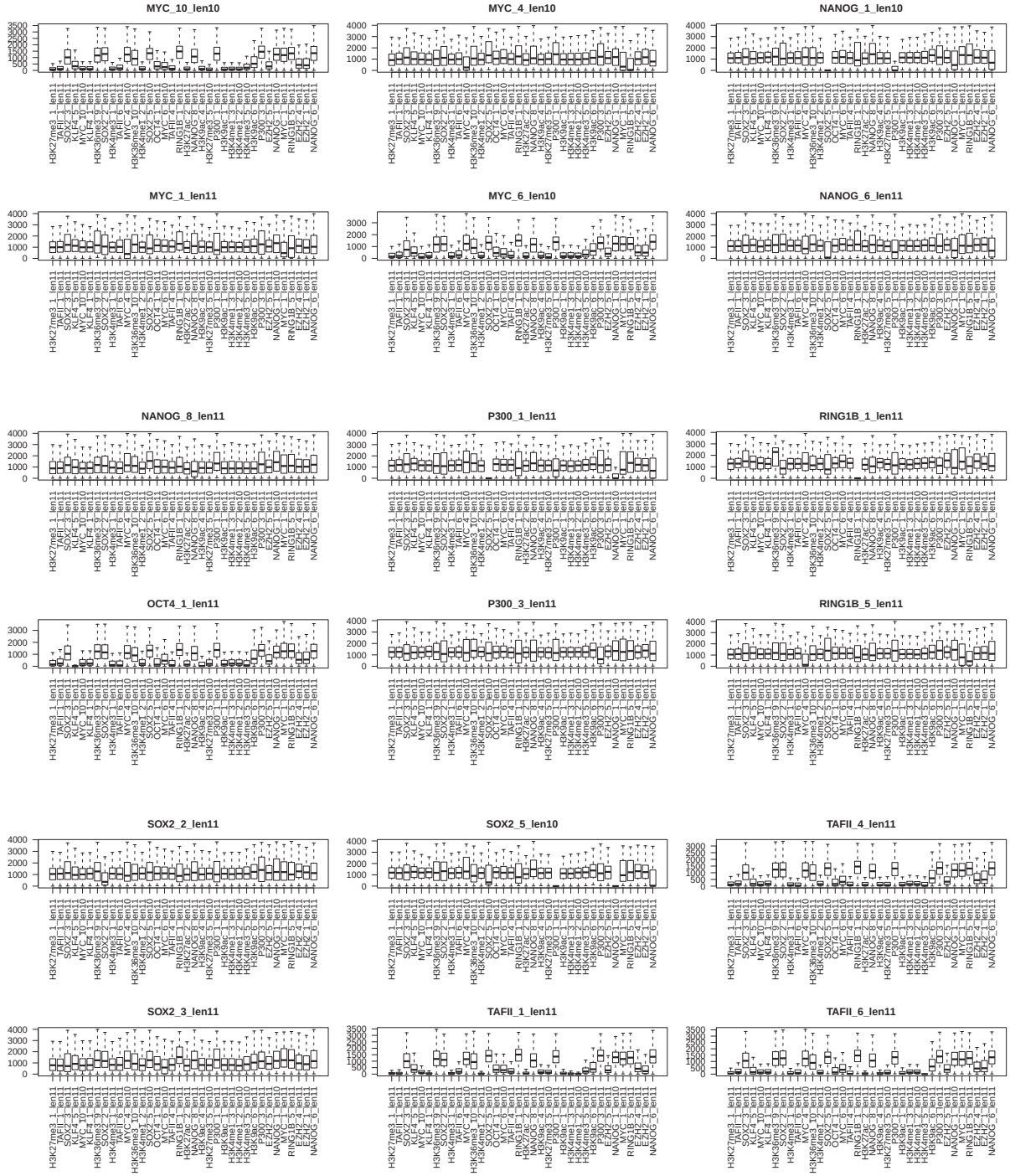


Figure S19. Distribution of the shortest distances for each motif-pair in the promoter's regions. For example, the first panel shows the distribution of the shortest genomic distances from other motifs to each EZH2_1_len11 motif site within the 4kb maximum range. The distribution of the shortest distance for two different instances of the EZH2_1_len11 motif is also presented. The plot is illustrated by the R 'boxplot' function with default parameters and 'outline = F'. The plots in other panels are similar. See Supplementary Methods section "Estimating the significance of motif networks by motif-motif proximity" for more details.

Figure S20

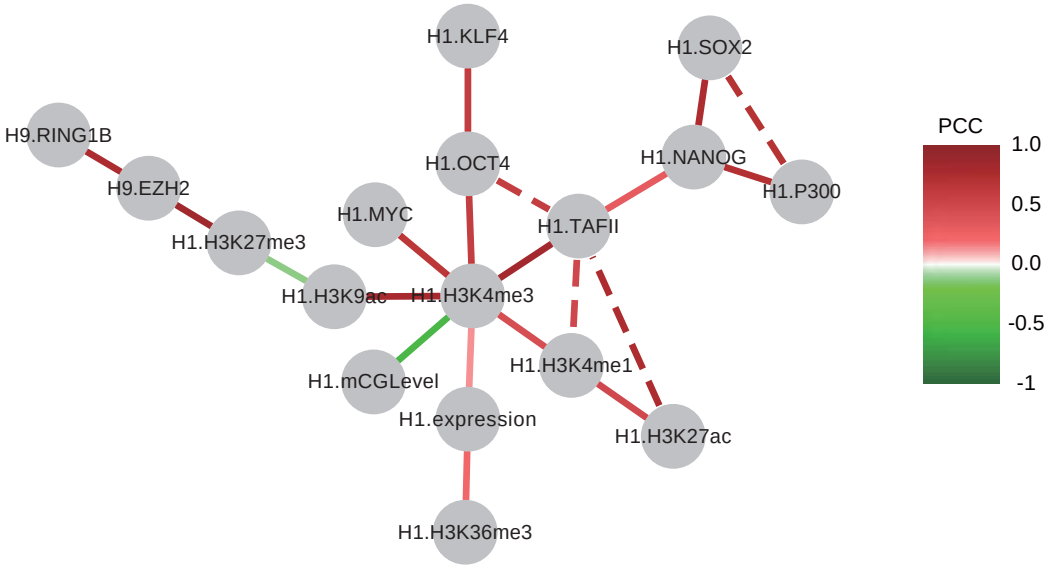


Figure S20. Four feedback edges (shown as dashed lines) inferred by SeqSpider on top of the hESC regulator network. All edges in this figure are colored according to the color legend based on the Pearson correlation coefficient (PCC) of the total tag counts in the TSS+/-2Kb region (or TTS+/-2Kb region for H3K36me3) of the two interacting nodes.

Figure S21

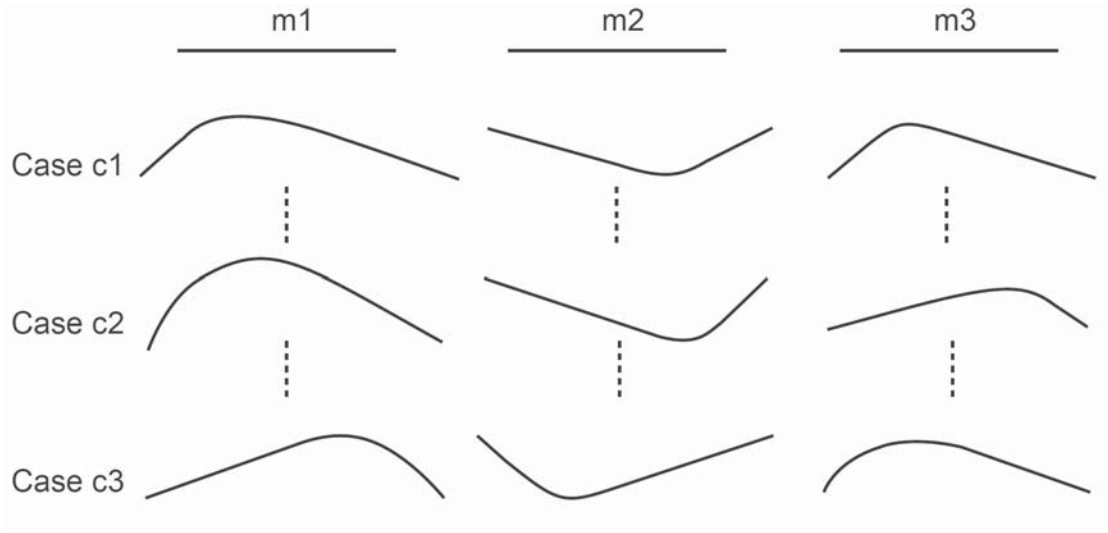


Figure S21. Covariations between modification profiles implicate potential interactions between corresponding regulatory factors. The profiles of modification m1, m2 and m3 at three training cases (genes or cluster centers) c1, c2 and c3 are illustrated. For m1 and m2, the patterns at c1 and c2 are much similar than the pattern at c3; while for m3, the pattern at c1 and c3 are much similar than c2, which suggests the covariation between m1 and m2 are better manifested in the three training cases. If this is true for a large number of training cases, it is highly likely that there is a potential interaction between m1 and m2. The similarity between tag profiles is captured by the proposed L1-RPS kernel.

Figure S22

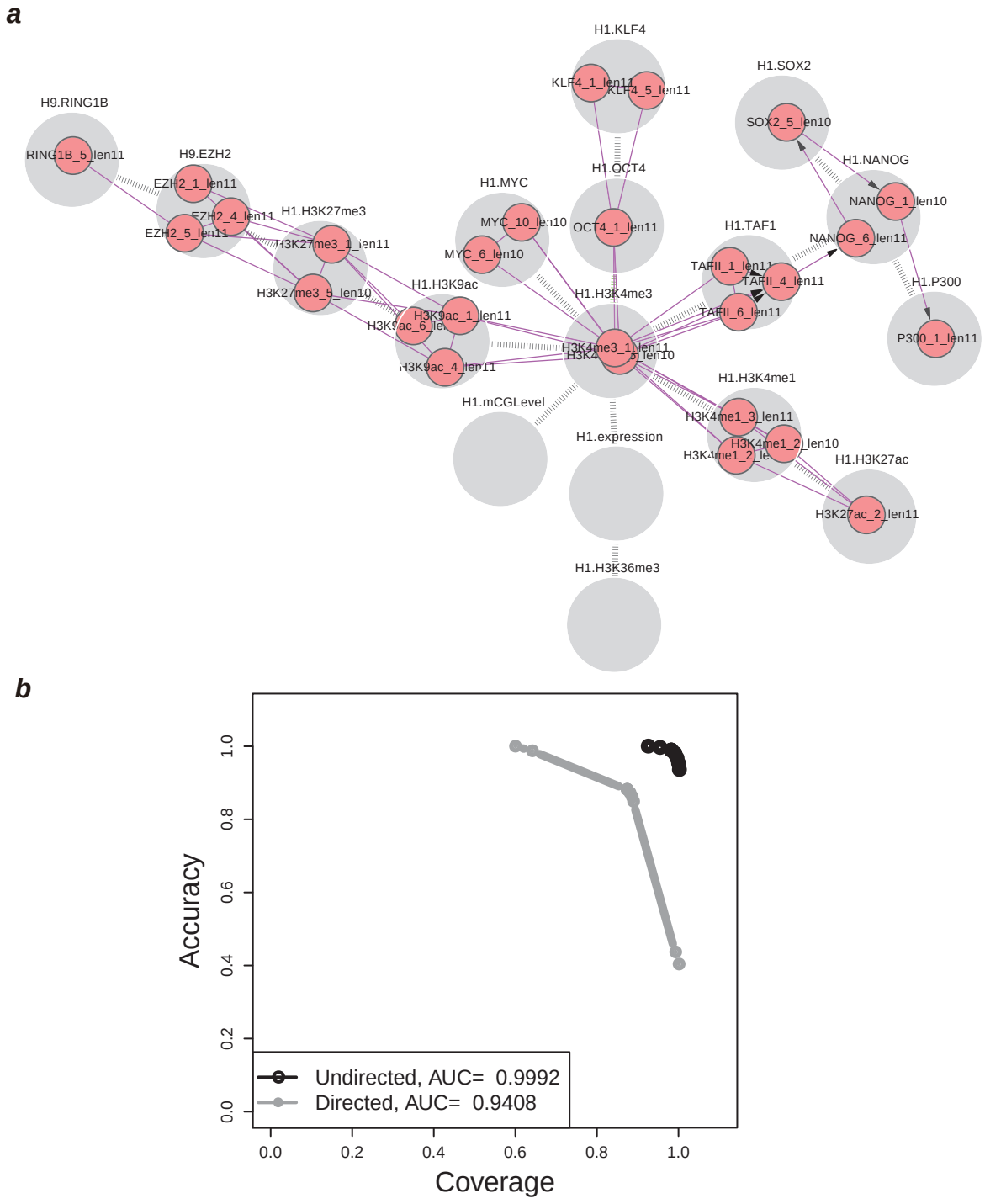


Figure S22. The constrained motif interaction network learned by SeqSpider algorithm. (a) Integrating the hESC regulator network (Figure 1A) and the motif distributions over all the genes in the genome results in a constrained, context-dependent motif interaction network to explain the sequence motifs that mediate the regulator interactions. Edges that appear in ≥ 7 PDAGs in the 10-fold data re-sampling based network learning procedure are selected to constitute the consensus network. Each motif name is composed of three components, the first is the name of the corresponding hESC regulator, the second is the rank of the motif in the DME2 output and the last is the length of the motif itself. (b) Directed / undirected edge network stability curves for the PDAG structures in the network learning procedure.

Figure S23

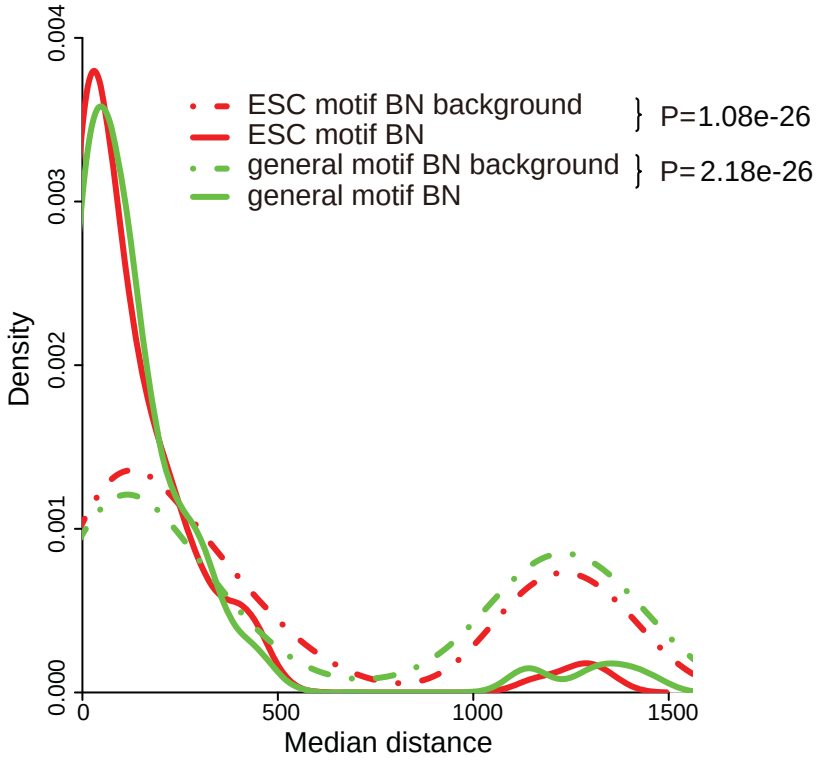


Figure S23. The distribution of genomic distances between motif interactors in the general (unconstrained, see Figure S11) / hESC context-specific (constrained, see Figure S22a) motif-motif interaction network compared with all possible pair-wise distances among nodes in the corresponding network. One-tailed Student's t-test is used to estimate the significance of the differences of the real versus the corresponding background distributions.

Figure S24

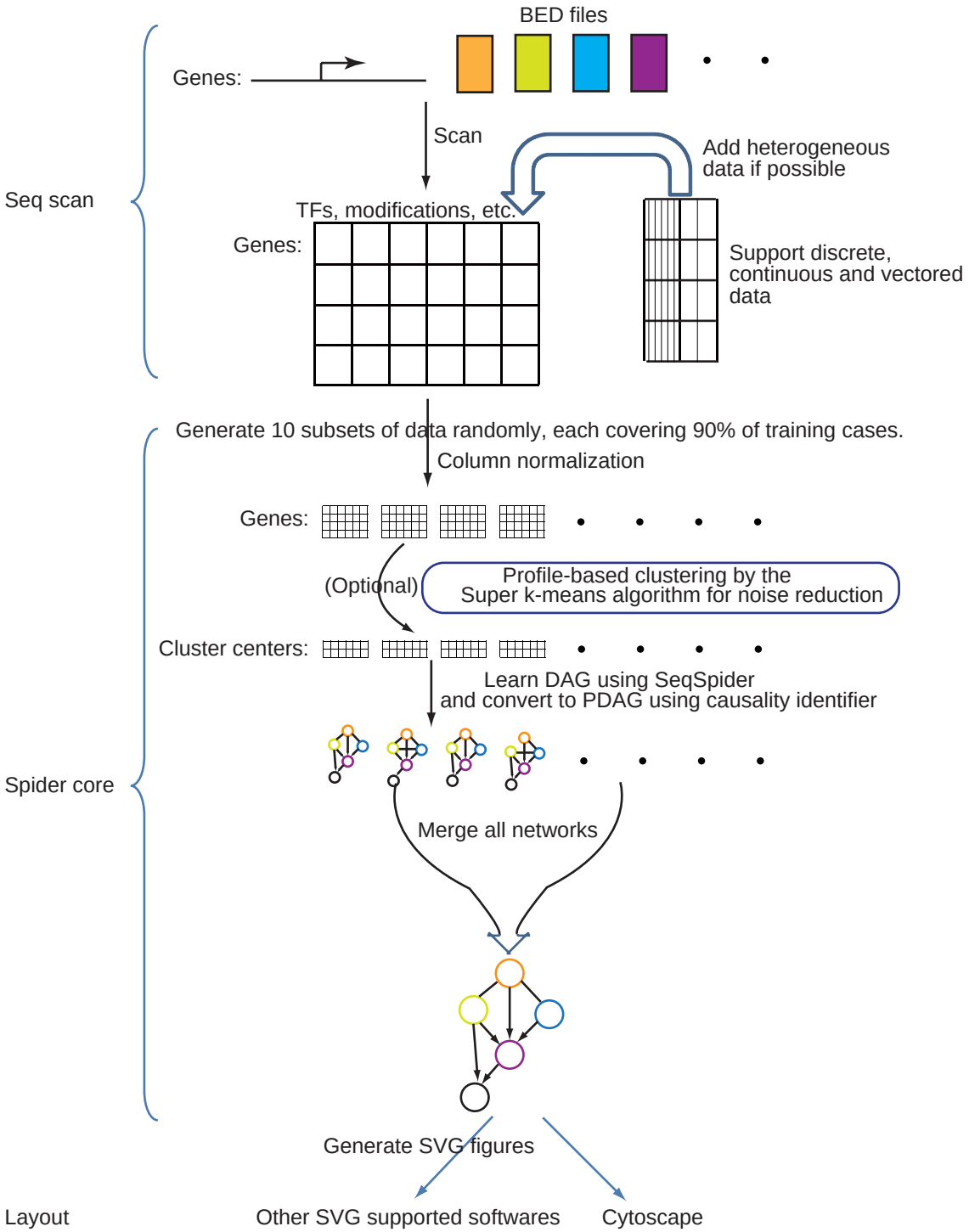
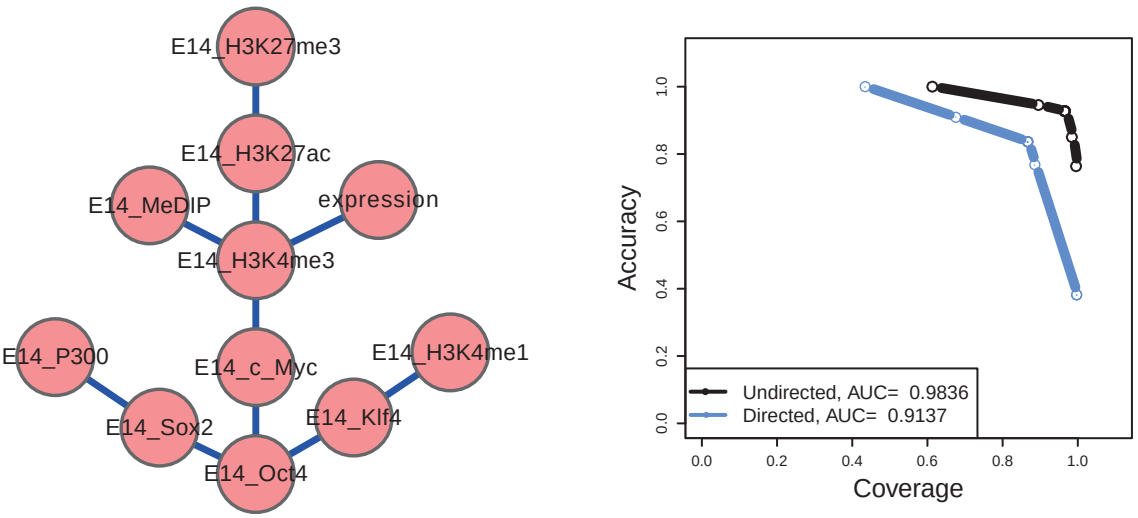

BED files
Genes:
Scan
Add heterogeneous data if possible
Seq scan
TFs, modifications, etc.
Genes:
Support discrete, continuous and vectored data
Generate 10 subsets of data randomly, each covering 90% of training cases.
Column normalization
Genes:
(Optional)
Profile-based clustering by the Super k-means algorithm for noise reduction
Cluster centers:
Learn DAG using SeqSpider and convert to PDAG using causality identifier
Spider core
Merge all networks
Generate SVG figures
Layout
Other SVG supported softwares
Cytoscape

Figure S24. The basic functional modules in SeqSpider. To facilitate researchers using this BN structure inference algorithm, we have packaged not only the algorithm itself, but also a number of useful tools for curating training data, inferring robust consensus network, making causal interpretation and visualizing the learned network structure into a coherent pipeline for analyzing deep sequencing data. Basically, these utilities are organized into three major functional modules: (a) the "Seq scan" module, which is used to parse deep sequencing data and prepare the gene-wise training data for the BN inference algorithm; (b) the "Spider core" module, which executes the SeqSpider algorithm in a data re-sampling based network learning setting to extract the consensus regulatory network; (c) the "layout" module, which converts the inferred network structure into the SVG format used in Cytoscape and other network visualization software (See Supplementary Methods for more details). Here, in the "Spider core" module, "causality identifier" denotes the algorithmic procedure for converting a DAG to a PDAG (implementing Meek's algorithm [17]), which is necessary for correctly identifying irreversible causal relationships from a BN.

Figure S25

a



b

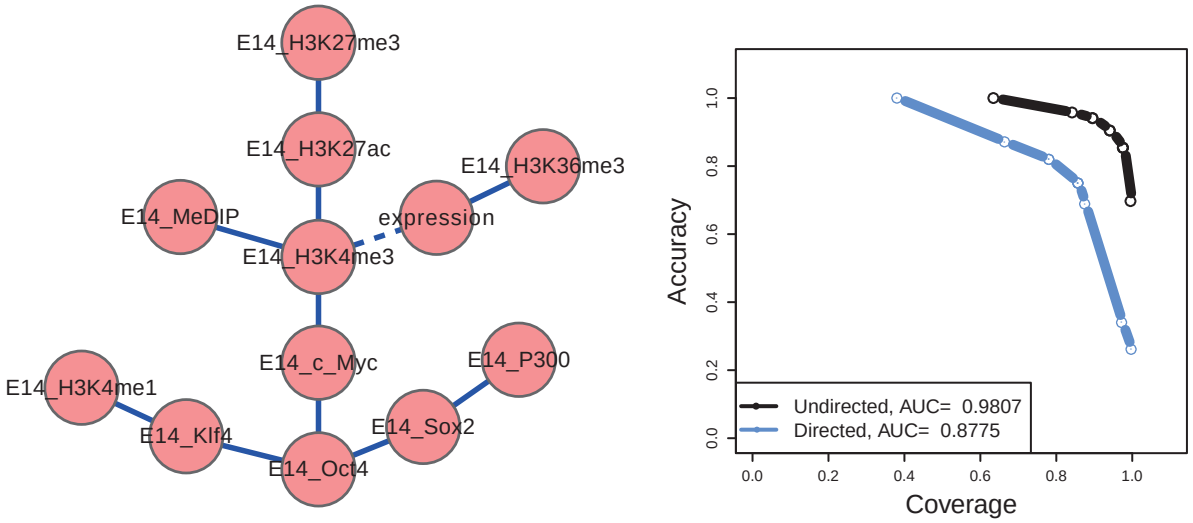


Figure S25. The regulator network learned from heterogeneous deep sequencing data of mouse embryonic stem cells (mESCs). (**a**) The 13-nodes consensus mESC network inferred from the tag distribution signal of TSS [-2kb, +2kb] regions, where H3K36me3 was not included in the network. The directed/undirected edge network stability curve for inferring this consensus network is also plotted. (**b**) The 14-nodes consensus mESC network inferred from TSS [-2kb, +2kb] signal of the 13 nodes in (a) and the TTS [-2kb, +2kb] signal for H3K36me3 (shown by solid lines). Again, the network stability curve is plotted with the mESC network. The post-BN feedback edge search algorithm uncovered on the 10-fold data an additional edge (highlighted by a dashed line), H3K4me3-gene expression, in more than 50% of the runs.

Supplementary Tables

Table S1. A summary of the functions enabled by the SeqSpider algorithm.

Type	Detail	Traditional BN*	SeqSpider#
Data (node)	Discretized	+	+
	Continuous	Limited	+
	Vector/Profiles#	-	+
Experiments	Single	+	+
	Multiple	-	+
	Noise removal	-	+
Reliability	Robustness	Low	High

* + indicates “capable of” while “-” means “unable to”.

or a hybrid of the three types of data above.

Table S2. The ChIP-Seq/BS-Seq/RNA-Seq datasets used in this study.

Dataset type	Details	Data type	Cell line	Laboratory
DNA methylation	DNA methylation	vector; real value	hES, H1	Joseph Ecker
Histone modifications	H3K27ac, H3K27me3, H3K36me3, H3K4me1, H3K4me3, H3K9ac, H3K9me3	vector; real value	hES, H1	Bing Ren
Gene expression	RNA-seq data	real value	hES, H1	Bing Ren
Transcript factor	OCT4, KLF, MYC, TAFII, P300, SOX2, NANOG	vector; real value	hES, H1	Bing Ren
PRC complex	EZH2 and RING1B	vector; real value	hES, H9	Tarjei Mikkelsen

Table S3. Comparing hESC regulator networks derived from profiled training data with different number of bins. Following the data curation/normalization and the data re-sampling procedure for deriving the hESC regulator network in Figure 1A, we test the effect on network inference when different numbers of bins are used to represent tag profiles in the [-2k, +2k] region surrounding TSSs (or TTS for H3K36me3). A comparison of the consensus networks derived from 6, 8, 12 or 14-bin profiles (denoted by network *B*) with the default network (denoted by network *A*) in Figure 1A (with 10-bin profiles) is shown in the table. Two indicators, the Dice's Coefficient *D* and the Jaccard index *J* are used to access the network similarities. Intermediate values for computing the two indices, such as the number of edges for the two networks and for their intersection and union, are also listed in the table.

Network *A*	**Network** *B*	**| A |**	**| B |**	**| A ∩ B |**	**| A ∪ B |**	**D**	**J**
10 bins	6 bins	16	15	14	17	0.903	0.824
10 bins	8 bins	16	14	14	16	0.933	0.875
10 bins	12 bins	16	13	13	16	0.897	0.813
10 bins	14 bins	16	14	13	17	0.867	0.765

Table S4. KGV scores for BNs on disjoint training / testing datasets. To test the generalization performance of the SeqSpider algorithm, we randomly divide the profiled hESC data at the gene level into equal-sized training and test sets and represent the data in each set by 1000 Super k-means cluster centers. A BN is derived from the training set and then its un-regularized KGV scores (panel a) and regularized KGV scores ($\lambda = 3.0$, panel b) are evaluated on both the training set and the test set. The within-sample / out-of-sample pair of KGV scores of 10 trials are listed in the table. See Note 12 for more details.

a)

Train	7743.05	6991.99	7914.47	7683.79	7503.18	7628.77	7672.63	7847.53	8029.69	7839.50
Test	7776.79	7015.60	8187.23	7350.48	7672.02	7817.95	7664.82	7949.25	7663.87	7664.62

b)

Train	3126.60	2890.06	3350.47	3339.59	3129.64	3147.29	3213.73	3312.80	3542.84	3355.92
Test	3105.12	2803.63	3623.01	2957.54	3260.62	3236.86	3165.75	3346.43	3103.59	3182.81

Table S5. The consistency of BN structures derived from disjoint sets of training data. Two BNs (denoted by A and B) are inferred separately on the disjoint hESC training / testing datasets (each represented by 1000 cluster centers). Then, the structural consistency of the two BN structures is measured by the Dice's Coefficient (D) and the Jaccard index (J) on their PDAG representations, which are calculated from the number of edges of the two PDAG themselves, their intersection and union ($|A|, |B|, |A \cap B|, |A \cup B|$). Documented in this table are the similarity scores resulting from 10 trials of the process above. See Note 12 for more details.

$\|A\|$	15	15	15	15	15	15	14	15	15	15
$\|B\|$	15	14	15	14	15	15	15	15	15	15
$\|A \cap B\|$	14	13	15	13	14	14	14	14	13	14
$\|A \cup B\|$	16	16	15	16	16	16	15	16	17	16
D	0.93	0.90	1.00	0.90	0.93	0.93	0.97	0.93	0.87	0.93
J	0.88	0.81	1.00	0.81	0.88	0.88	0.93	0.88	0.76	0.88

Table S6. The (regularized) KGV scores for the 4 known interactions not in the hESC regulator network in Figure 1A (panel **(a)**) and the 16 edges in the network (panel **(b)**). The weight of the complexity term in the BN scoring function is set to $\lambda = 3.0$.

a)

Edge	KGV score
H1.NANOG -- H1.OCT4	-224.297309
H1.SOX2 -- H1.OCT4	-269.076411
H1.P300 -- H1.SOX2	308.842062
H1.P300 -- H1.OCT4	-277.743219

b)

Edge	KGV score
H1.H3K4me3 -- H1.H3K9ac	354.024763
H1.KLF4 -- H1.OCT4	267.14138
H1.H3K4me3 -- H1.expression	213.858355
H1.H3K27ac -- H1.H3K4me1	362.715421
H1.H3K27me3 -- H9.EZH2	222.302419
H1.H3K27me3 -- H1.H3K9ac	290.933779
H9.EZH2 -- H9.RING1B	66.17088
H1.NANOG -- H1.P300	346.09213
H1.NANOG -- H1.SOX2	482.278844
H1.H3K4me3 -- H1.OCT4	410.551977
H1.H3K4me1 -- H1.H3K4me3	432.554735
H1.H3K36me3 -- H1.expression	237.596797
H1.H3K4me3 -- H1.TAFII	430.411803
H1.NANOG -- H1.TAFII	20.739222
H1.H3K4me3 -- H1.MYC	363.714595
H1.H3K4me3 -- H1.mCGLevel	112.052818

Table S7. The total number of cDNA short reads for each chromatin modification / transcription factor. Typically, epigenetic modification peaks occupy more or longer regions in the genome than transcription factor binding sites. As a result, in general modification data require much more reads to achieve sufficient sequencing depth, especially for those with block-like tag distributions, such as H3K9me3 and H4K20me3. However, compared to other modifications, H3K9me3 has the smallest number of reads (approximately only half of the second smallest factor: H3K27ac). Therefore, it is likely that H3K9me3 enriched regions are not sufficiently sequenced (See Figure S16 and Table S8 and for more evidence that supports this argument).

name	reads counts
H3K36me3	41934237
H3K27me3	35823177
H3K4me1	19048630
H3K4me3	13252801
H3K9ac	12056771
H3K27ac	10725669
EZH2	6257563
H3K9me3	5943995
TAFII	5448239
NANOG	5082225
SOX2	3560183
P300	1888673
MYC	1676054
OCT4	1469135
RING1B	1460846
KLF4	462762

Table S8. The total number of peak regions detected by SICER / MACS for each epigenetic modification / transcription factor (using P-value cutoff 1e-5). Clearly, H3K9me3 has the least number of peaks among all epigenetic modifications (which is even smaller than half of the second smallest factor: H3K9ac), suggesting the sequencing depth of H3K9me3 is probably not enough (See Figure S16 and Table S7 for more discussions).

name	peak counts
H3K4me1	44579
H3K36me3	44049
NANOG	33006
H3K27ac	27950
H3K4me3	18827
TAFII	16570
SOX2	15405
H3K27me3	14946
H3K9ac	11014
P300	6022
OCT4	5107
H3K9me3	4695
KLF4	2539
MYC	2043
EZH2	1899
RING1B	35

Table S9. Four feedback edges identified on top of the hESC regulator network in Figure 1A and their numbers of occurrence. See Note 19 for more details.

Edge	Numbers of occurrence in 50 runs
H1.H3K4me1- H1.TAFII	20
H1.OCT4 - H1.TAFII	14
H1.P300 - H1.SOX2	12
H1.H3K27ac - H1.TAFII	7

Table S10. Overlap of the networks learned from randomly selected genes and the consensus hESC regulator network. **(a)** Network *A* is the network learned from 10024 genes (the same number as Cluster 2 in Figure S14a) randomly selected from the genome without replacement. Network *B* is the consensus hESC network in Figure 1A. The degree of H3K4me3 for the two networks and two network similarity indices (Dice's coefficient (D) and Jaccard index (J)) are listed. **(b)** Similar to (a), but only 40% genes randomly selected from the genome are used to infer Network *A*.

a)

| A |	| B |	| A ∩ B |	| A ∪ B |	Degree of H3K4me3 in A	Degree of H3K4me3 in B	J	D
15	16	13	18	7	7	0.72	0.84
14	16	13	17	5	7	0.76	0.87
15	16	15	16	7	7	0.94	0.97
15	16	15	16	7	7	0.94	0.97
15	16	13	18	7	7	0.72	0.84
15	16	15	16	7	7	0.94	0.97
14	16	14	16	7	7	0.88	0.93
15	16	15	16	7	7	0.94	0.97
14	16	13	17	8	7	0.76	0.87
15	16	14	17	6	7	0.82	0.90

b)

| A |	| B |	| A ∩ B |	| A ∪ B |	Degree of H3K4me3 in A	Degree of H3K4me3 in B	J	D
14	16	12	18	8	7	0.67	0.80
14	16	14	16	7	7	0.88	0.93
14	16	14	16	7	7	0.88	0.93
13	16	13	16	6	7	0.81	0.90
13	16	13	16	6	7	0.81	0.90
15	16	14	17	6	7	0.82	0.90
13	16	13	16	6	7	0.81	0.90
14	16	14	16	7	7	0.88	0.93
14	16	13	17	6	7	0.76	0.87
15	16	15	16	7	7	0.94	0.97

Supplementary Datasets

Dataset S1. The position weight matrices of the motifs detected from the peak regions of the ChIP-Seq signals by DME2 (also shown in Figure S15).

AC EZH2_4_len11
XX
TY Motif
XX
ID CCSRASMYCKS
XX

P0	A	C	G	T
01	0	139	0	0
02	0	139	0	0
03	0	78	61	0
04	88	0	51	0
05	139	0	0	0
06	0	79	60	0
07	52	87	0	0
08	0	84	0	55
09	0	139	0	0
10	0	0	67	72
11	0	85	54	0

XX
//
AC EZH2_1_len11
XX
TY Motif
XX
ID GBAGCAKCMKC
XX

P0	A	C	G	T
01	0	0	97	0
02	0	81	11	5
03	97	0	0	0
04	0	0	97	0
05	0	97	0	0
06	97	0	0	0
07	0	0	86	11
08	0	97	0	0
09	74	23	0	0
10	0	0	78	19
11	0	97	0	0

XX
AT BGCOUNT=124
AT CORRECTEDBGCOUNT=62
AT FGCOUNT=97
AT INFO=1.74482
AT SCORE=321.536
XX
//
AC EZH2_5_len11
XX
TY Motif
XX
ID SSCGARCWBMG
XX

P0	A	C	G	T
01	0	50	27	0
02	0	51	26	0
03	0	77	0	0
04	0	0	77	0
05	77	0	0	0
06	32	0	45	0
07	0	77	0	0
08	46	0	0	31
09	0	23	31	23
10	29	48	0	0
11	0	0	77	0

XX
//
AC H3K27ac_2_len11
XX
TY Motif
XX
ID SCCGCSSSYNS
XX

P0	A	C	G	T
01	0	287	112	0
02	0	399	0	0
03	0	399	0	0
04	0	0	399	0
05	0	399	0	0
06	0	306	93	0
07	0	241	158	0
08	0	340	59	0
09	0	344	0	55

10 46 92 216 45
11 0 292 107 0
XX
AT BGCOUNT=279
AT CORRECTEDBGCOUNT=140
AT FGCOUNT=399
AT INFO=1.59895
AT SCORE=4093.1
XX
//
AC H3K27me3_5_len10
XX
TY Motif
XX
ID SSSSGCGMCS
XX
P0 A C G T
01 0 315 124 0
02 0 362 77 0
03 0 277 162 0
04 0 285 154 0
05 0 0 439 0
06 0 439 0 0
07 0 0 439 0
08 42 397 0 0
09 0 439 0 0
10 0 349 90 0
XX
AT BGCOUNT=354
AT CORRECTEDBGCOUNT=179
AT FGCOUNT=439
AT INFO=1.65911
AT SCORE=4234.38
XX
//
AC H3K27me3_1_len11
XX
TY Motif
XX
ID SCSSSSGCCSS
XX
P0 A C G T
01 0 144 553 0
02 0 697 0 0

03 0 565 132 0
04 0 150 547 0
05 0 529 168 0
06 0 595 102 0
07 0 0 697 0
08 0 697 0 0
09 0 697 0 0
10 0 165 532 0
11 0 612 85 0
XX
AT BGCOUNT=554
AT CORRECTEDBGCOUNT=280
AT FGCOUNT=697
AT INFO=1.58831
AT SCORE=4245.93
XX
//
AC H3K36me3_9_len11
XX
TY Motif
XX
ID ACCAGTSAGAC
XX
P0 A C G T
01 8 0 0 0
02 0 8 0 0
03 0 8 0 0
04 8 0 0 0
05 0 0 8 0
06 0 0 0 8
07 0 3 5 0
08 8 0 0 0
09 0 0 8 0
10 8 0 0 0
11 0 8 0 0
XX
AT BGCOUNT=11
AT CORRECTEDBGCOUNT=5
AT FGCOUNT=8
AT INFO=1.96937
AT SCORE=59.4128
XX
//
AC H3K36me3_10_len11

XX
TY Motif
XX
ID CAGGCAGCTCA
XX

P0	A	C	G	T
01	0	20	0	0
02	20	0	0	0
03	0	0	20	0
04	0	0	20	0
05	0	20	0	0
06	20	0	0	0
07	0	0	20	0
08	0	20	0	0
09	0	0	0	20
10	0	20	0	0
11	20	0	0	0

XX
AT BGCOUNT=23
AT CORRECTEDBGCOUNT=11
AT FGCOUNT=20
AT INFO=2.05825
AT SCORE=-104.779
XX
//
AC H3K4me1_2_len10
XX
TY Motif
XX
ID DSCBGCGGCG
XX

P0	A	C	G	T
01	16	0	102	10
02	0	25	103	0
03	0	128	0	0
04	0	18	81	29
05	0	0	128	0
06	0	128	0	0
07	0	0	128	0
08	0	0	128	0
09	0	128	0	0
10	0	0	128	0

XX
AT BGCOUNT=148

AT CORRECTEDBGCOUNT=74
AT FGCOUNT=128
AT INFO=1.8012
AT SCORE=814.701
XX
//
AC H3K4me1_2_len11
XX
TY Motif
XX
ID BGMSGCSGCGR
XX

P0	A	C	G	T
01	0	14	101	8
02	0	0	123	0
03	30	93	0	0
04	0	13	110	0
05	0	0	123	0
06	0	123	0	0
07	0	16	107	0
08	0	0	123	0
09	0	123	0	0
10	0	0	123	0
11	15	0	108	0

XX
AT BGCOUNT=136
AT CORRECTEDBGCOUNT=68
AT FGCOUNT=123
AT INFO=1.75716
AT SCORE=743.015
XX
//
AC H3K4me1_3_len11
XX
TY Motif
XX
ID CGGCGBBRGMB
XX

P0	A	C	G	T
01	0	146	0	0
02	0	0	146	0
03	11	0	135	0
04	0	146	0	0
05	0	0	146	0

06 0 32 99 15
07 0 81 50 15
08 33 0 113 0
09 0 0 146 0
10 31 115 0 0
11 0 27 97 22
XX
AT BGCOUNT=155
AT CORRECTEDBGCOUNT=77
AT FGCOUNT=146
AT INFO=1.65067
AT SCORE=1022.52
XX
//
AC H3K4me3_5_len10
XX
TY Motif
XX
ID CCCCGSSSSB
XX
P0 A C G T
01 0 941 0 0
02 0 941 0 0
03 0 941 0 0
04 0 941 0 0
05 0 0 941 0
06 0 787 154 0
07 0 767 174 0
08 0 762 179 0
09 0 825 116 0
10 0 602 245 94
XX
AT BGCOUNT=400
AT CORRECTEDBGCOUNT=200
AT FGCOUNT=941
AT INFO=1.6315
AT SCORE=9402.21
XX
//
AC H3K4me3_10_len10
XX
TY Motif
XX
ID SGVGGAGSVG

XX
P0 A C G T
01 0 147 694 0
02 0 0 841 0
03 505 112 224 0
04 0 0 841 0
05 0 0 841 0
06 841 0 0 0
07 0 0 841 0
08 0 128 713 0
09 332 107 402 0
10 0 0 841 0
XX
AT BGCOUNT=859
AT CORRECTEDBGCOUNT=431
AT FGCOUNT=841
AT INFO=1.61314
AT SCORE=5327.8
XX
//
AC H3K4me3_1_len11
XX
TY Motif
XX
ID CSSSSSSCGCS
XX
P0 A C G T
01 0 1059 0 0
02 0 882 177 0
03 0 157 902 0
04 0 820 239 0
05 0 838 221 0
06 0 320 739 0
07 0 924 135 0
08 0 1059 0 0
09 0 0 1059 0
10 0 1059 0 0
11 0 938 121 0
XX
AT BGCOUNT=243
AT CORRECTEDBGCOUNT=121
AT FGCOUNT=1059
AT INFO=1.5638
AT SCORE=10377.2

XX
//
AC H3K9ac_1_len11
XX
TY Motif
XX
ID SMGGSGSSGSS
XX

P0	A	C	G	T
01	0	135	885	0
02	172	848	0	0
03	0	0	1020	0
04	0	0	1020	0
05	0	718	302	0
06	0	0	1020	0
07	0	236	784	0
08	0	850	170	0
09	0	0	1020	0
10	0	153	867	0
11	0	689	331	0

XX
AT BGCOUNT=297
AT CORRECTEDBGCOUNT=149
AT FGCOUNT=1020
AT INFO=1.55148
AT SCORE=9098.86
XX
//
AC H3K9ac_6_len11
XX
TY Motif
XX
ID GMGSRGSMSGM
XX

P0	A	C	G	T
01	0	0	626	0
02	398	228	0	0
03	0	0	626	0
04	0	159	467	0
05	384	0	242	0
06	0	0	626	0
07	0	168	458	0
08	368	258	0	0
09	0	112	514	0

10 0 0 626 0
11 370 256 0 0
XX
AT BGCOUNT=399
AT CORRECTEDBGCOUNT=200
AT FGCOUNT=626
AT INFO=1.56344
AT SCORE=5308.07
XX
//
AC H3K9ac_4_len11
XX
TY Motif
XX
ID SSSSCSGGGMS
XX
P0 A C G T
01 0 147 574 0
02 0 187 534 0
03 0 164 557 0
04 0 185 536 0
05 0 721 0 0
06 0 199 522 0
07 0 0 721 0
08 0 0 721 0
09 0 0 721 0
10 185 536 0 0
11 0 300 421 0
XX
AT BGCOUNT=287
AT CORRECTEDBGCOUNT=144
AT FGCOUNT=721
AT INFO=1.56575
AT SCORE=8545.37
XX
//
AC KLF4_1_len11
XX
TY Motif
XX
ID KYBGCBSCGGC
XX
P0 A C G T
01 0 0 101 13

02 0 97 0 17
03 0 10 94 10
04 0 0 114 0
05 0 114 0 0
06 0 20 85 9
07 0 17 97 0
08 0 114 0 0
09 0 0 114 0
10 0 0 114 0
11 0 114 0 0
XX
AT BGCOUNT=113
AT CORRECTEDBGCOUNT=56
AT FGCOUNT=114
AT INFO=1.59807
AT SCORE=592.376
XX
//
AC KLF4_5_len11
XX
TY Motif
XX
ID GGHGGGGMGNS
XX
P0 A C G T
01 0 0 182 0
02 0 0 182 0
03 90 58 0 34
04 0 0 182 0
05 0 0 182 0
06 0 0 182 0
07 0 0 182 0
08 115 67 0 0
09 0 0 182 0
10 32 24 115 11
11 0 30 152 0
XX
AT BGCOUNT=152
AT CORRECTEDBGCOUNT=76
AT FGCOUNT=182
AT INFO=1.55297
AT SCORE=1233.57
XX
//

AC MYC_10_len10
XX
TY Motif
XX
ID SMGCGCCSCS
XX

P0	A	C	G	T
01	0	64	35	0
02	40	59	0	0
03	0	0	99	0
04	0	99	0	0
05	0	0	99	0
06	0	99	0	0
07	0	99	0	0
08	0	41	58	0
09	0	99	0	0
10	0	64	35	0

XX
AT BGCOUNT=96
AT CORRECTEDBGCOUNT=48
AT FGCOUNT=99
AT INFO=1.61144
AT SCORE=740.492
XX
//
AC MYC_6_len10
XX
TY Motif
XX
ID CMGDGSGCRC
XX

P0	A	C	G	T
01	0	69	0	0
02	46	23	0	0
03	0	0	69	0
04	21	0	38	10
05	0	0	69	0
06	0	23	46	0
07	0	0	69	0
08	0	69	0	0
09	21	0	48	0
10	0	69	0	0

XX
AT BGCOUNT=53

AT CORRECTEDBGCOUNT=26
AT FGCOUNT=69
AT INFO=1.6071
AT SCORE=674.676
XX
//
AC MYC_4_len10
XX
TY Motif
XX
ID CRCAACDCWM
XX

P0	A	C	G	T
01	0	37	0	0
02	34	0	3	0
03	0	37	0	0
04	37	0	0	0
05	37	0	0	0
06	0	37	0	0
07	32	0	2	3
08	0	37	0	0
09	33	0	0	4
10	33	4	0	0

XX
AT BGCOUNT=15
AT CORRECTEDBGCOUNT=7
AT FGCOUNT=37
AT INFO=1.66703
AT SCORE=174.794
XX
//
AC MYC_1_len11
XX
TY Motif
XX
ID CMAMAHDASAC
XX

P0	A	C	G	T
01	0	39	0	0
02	35	4	0	0
03	39	0	0	0
04	6	33	0	0
05	39	0	0	0
06	4	30	0	5

07 30 0 4 5
08 39 0 0 0
09 0 34 5 0
10 39 0 0 0
11 0 39 0 0
XX
AT BGCOUNT=11
AT CORRECTEDBGCOUNT=5
AT FGCOUNT=39
AT INFO=1.56992
AT SCORE=215.762
XX
//
AC NANOG_1_len10
XX
TY Motif
XX
ID VMATTTRCAT
XX
P0 A C G T
01 44 25 26 0
02 55 40 0 0
03 95 0 0 0
04 0 0 0 95
05 0 0 0 95
06 0 0 0 95
07 36 0 59 0
08 0 95 0 0
09 95 0 0 0
10 0 0 0 95
XX
AT BGCOUNT=86
AT CORRECTEDBGCOUNT=43
AT FGCOUNT=95
AT INFO=1.60178
AT SCORE=806.133
XX
//
AC NANOG_6_len11
XX
TY Motif
XX
ID YTSAAWKGCWA
XX

P0	A	C	G	T
01	0	13	0	35
02	0	0	0	48
03	0	26	22	0
04	48	0	0	0
05	48	0	0	0
06	37	0	0	11
07	0	0	14	34
08	0	0	48	0
09	0	48	0	0
10	36	0	0	12
11	48	0	0	0

XX
AT BGCOUNT=44
AT CORRECTEDBGCOUNT=22
AT FGCOUNT=48
AT INFO=1.60265
AT SCORE=455.429
XX
//
AC NANOG_8_len11
XX
TY Motif
XX
ID RGAMAAGARKG
XX

P0	A	C	G	T
01	19	0	7	0
02	0	0	26	0
03	26	0	0	0
04	21	5	0	0
05	26	0	0	0
06	26	0	0	0
07	0	0	26	0
08	26	0	0	0
09	19	0	7	0
10	0	0	11	15
11	0	0	26	0

XX
AT BGCOUNT=25
AT CORRECTEDBGCOUNT=12
AT FGCOUNT=26
AT INFO=1.6847
AT SCORE=249.51

XX
//
AC OCT4_1_len11
XX
TY Motif
XX
ID YCKCCCYDCSS
XX

P0	A	C	G	T
01	0	198	0	40
02	0	238	0	0
03	0	0	88	150
04	0	238	0	0
05	0	238	0	0
06	0	238	0	0
07	0	182	0	56
08	32	0	111	95
09	0	238	0	0
10	0	200	38	0
11	0	202	36	0

XX
AT BGCOUNT=156
AT CORRECTEDBGCOUNT=78
AT FGCOUNT=238
AT INFO=1.55132
AT SCORE=2109.93
XX
//
AC P300_1_len11
XX
TY Motif
XX
ID SAVATGYWAAT
XX

P0	A	C	G	T
01	0	39	49	0
02	88	0	0	0
03	48	15	25	0
04	88	0	0	0
05	0	0	0	88
06	0	0	88	0
07	0	52	0	36
08	61	0	0	27
09	88	0	0	0

10 88 0 0 0
11 0 0 0 88
XX
AT BGCOUNT=59
AT CORRECTEDBGCOUNT=29
AT FGCOUNT=88
AT INFO=1.55178
AT SCORE=935.415
XX
//
AC P300_3_len11
XX
TY Motif
XX
ID MATGGAATYAK
XX
P0 A C G T
01 19 1 0 0
02 20 0 0 0
03 0 0 0 20
04 0 0 20 0
05 0 0 20 0
06 20 0 0 0
07 20 0 0 0
08 0 0 0 20
09 0 17 0 3
10 20 0 0 0
11 0 0 1 19
XX
AT BGCOUNT=24
AT CORRECTEDBGCOUNT=12
AT FGCOUNT=20
AT INFO=1.70505
AT SCORE=74.9732
XX
//
AC RING1B_1_len11
XX
TY Motif
XX
ID TGTATAWATAC
XX
P0 A C G T
01 0 0 0 16

02 0 0 16 0
03 0 0 0 16
04 16 0 0 0
05 0 0 0 16
06 16 0 0 0
07 1 0 0 15
08 16 0 0 0
09 0 0 0 16
10 16 0 0 0
11 0 16 0 0
XX
AT BGCOUNT=2
AT CORRECTEDBGCOUNT=1
AT FGCOUNT=16
AT INFO=1.86121
AT SCORE=30.7013
XX
//
AC RING1B_5_len11
XX
TY Motif
XX
ID TTGTGATGTGT
XX
P0 A C G T
01 0 0 0 14
02 0 0 0 14
03 0 0 14 0
04 0 0 0 14
05 0 0 14 0
06 14 0 0 0
07 0 0 0 14
08 0 0 14 0
09 0 0 0 14
10 0 0 14 0
11 0 0 0 14
XX
AT BGCOUNT=19
AT CORRECTEDBGCOUNT=9
AT FGCOUNT=14
AT INFO=1.98138
AT SCORE=21.7952
XX
//

AC SOX2_2_len11
XX
TY Motif
XX
ID NAACAAWAGRR
XX
P0 A C G T
01 31 7 17 9
02 64 0 0 0
03 64 0 0 0
04 0 64 0 0
05 64 0 0 0
06 64 0 0 0
07 50 0 0 14
08 64 0 0 0
09 0 0 64 0
10 38 0 26 0
11 48 0 16 0
XX
AT BGCOUNT=44
AT CORRECTEDBGCOUNT=22
AT FGCOUNT=64
AT INFO=1.57667
AT SCORE=684.45
XX
//
AC SOX2_3_len11
XX
TY Motif
XX
ID ABHAGRKGKCR
XX
P0 A C G T
01 124 0 0 0
02 0 69 44 11
03 28 39 0 57
04 124 0 0 0
05 0 0 124 0
06 60 0 64 0
07 0 0 63 61
08 0 0 124 0
09 0 0 95 29
10 0 124 0 0
11 98 0 26 0

XX
AT BGCOUNT=67
AT CORRECTEDBGCOUNT=33
AT FGCOUNT=124
AT INFO=1.5548
AT SCORE=1452.43
XX
//
AC SOX2_5_len10
XX
TY Motif
XX
ID AWATGCWAMY
XX

P0	A	C	G	T
01	86	0	0	0
02	59	0	0	27
03	86	0	0	0
04	0	0	0	86
05	0	0	86	0
06	0	86	0	0
07	58	0	0	28
08	86	0	0	0
09	73	13	0	0
10	0	24	0	62

XX
AT BGCOUNT=94
AT CORRECTEDBGCOUNT=47
AT FGCOUNT=86
AT INFO=1.60646
AT SCORE=591.935
XX
//
AC TAFII_4_len11
XX
TY Motif
XX
ID NGCCSCGCSYY
XX

P0	A	C	G	T
01	51	71	119	24
02	0	0	265	0
03	0	265	0	0
04	0	265	0	0

05 0 221 44 0
06 0 265 0 0
07 0 0 265 0
08 0 265 0 0
09 0 223 42 0
10 0 216 0 49
11 0 208 0 57
XX
AT BGCOUNT=73
AT CORRECTEDBGCOUNT=36
AT FGCOUNT=265
AT INFO=1.58587
AT SCORE=3769.45
XX
//
AC TAFII_6_len11
XX
TY Motif
XX
ID SCCCGCSSCSN
XX
P0 A C G T
01 0 208 64 0
02 0 272 0 0
03 0 272 0 0
04 0 272 0 0
05 0 0 272 0
06 0 272 0 0
07 0 206 66 0
08 0 200 72 0
09 0 272 0 0
10 0 171 101 0
11 37 58 118 59
XX
AT BGCOUNT=152
AT CORRECTEDBGCOUNT=76
AT FGCOUNT=272
AT INFO=1.55045
AT SCORE=3130.17
XX
//
AC TAFII_1_len11
XX
TY Motif

XX
ID GSNGSSGCYGC
XX
P0 A C G T
01 0 0 312 0
02 0 233 79 0
03 31 173 50 58
04 0 0 312 0
05 0 227 85 0
06 0 256 56 0
07 0 0 312 0
08 0 312 0 0
09 0 229 0 83
10 0 0 312 0
11 0 312 0 0
XX
AT BGCOUNT=189
AT CORRECTEDBGCOUNT=94
AT FGCOUNT=312
AT INFO=1.55688
AT SCORE=3026.54
XX
//

Dataset S2. Eight consensus hESC regulator networks derived from the 10-fold data re-sampling based network learning procedure with different types of training data / learning algorithms. vec/real/dis denotes vectored / real-valued / discrete training data; null/sk/ap/kmeans stands for no clustering / the Super k-means / affinity propagation / ordinary k-means algorithm-based profile-clustering was performed as a data preprocessing step.

ap.vec
H1.H3K4me3 -- H1.H3K9ac
H1.H3K4me1 -- H1.TAFII
H1.H3K27ac -- H1.H3K4me1
H1.H3K27me3 -- H1.H3K9ac
H1.H3K4me3 -- H1.OCT4
H1.H3K36me3 -- H1.expression
H1.H3K4me3 -- H1.expression
H1.H3K27me3 -- H9.EZH2
H1.H3K4me1 -- H1.H3K4me3

kmeans.vec
H1.H3K4me3 -- H1.H3K9ac
H1.NANOG -- H1.SOX2
H1.H3K27ac -- H1.H3K4me1
H1.H3K4me1 -- H1.H3K4me3
H1.H3K4me3 -- H1.TAFII
H1.H3K4me3 -- H1.mCGLevel
H1.H3K4me3 -- H1.OCT4

sk.vec
H1.H3K4me3 -- H1.H3K9ac
H1.KLF4 -- H1.OCT4
H1.H3K4me3 -- H1.expression
H1.H3K27ac -- H1.H3K4me1
H1.H3K27me3 -- H9.EZH2
H1.H3K27me3 -- H1.H3K9ac
H9.EZH2 -- H9.RING1B
H1.NANOG -- H1.P300
H1.NANOG -- H1.SOX2
H1.H3K4me3 -- H1.OCT4
H1.H3K4me1 -- H1.H3K4me3
H1.H3K36me3 -- H1.expression
H1.H3K4me3 -- H1.TAFII
H1.NANOG -- H1.TAFII

H1.H3K4me3 -- H1.MYC
H1.H3K4me3 -- H1.mCGLevel

sk.real
H9.EZH2 -> H1.OCT4
H9.EZH2 -> H9.RING1B
H9.EZH2 -> H1.mCGLevel
H1.H3K27ac -- H1.H3K9ac
H1.H3K27ac -- H1.expression
H1.H3K27ac -> H1.H3K4me1
H1.H3K27me3 -- H1.H3K9ac
H1.H3K27me3 -> H9.EZH2
H1.H3K27me3 -> H1.H3K4me1
H1.H3K36me3 -- H1.H3K9ac
H1.H3K4me1 -> H1.TAFII
H1.H3K9ac -> H1.TAFII
H1.H3K9ac -> H1.mCGLevel
H1.NANOG -> H9.RING1B
H1.OCT4 -> H1.KLF4
H1.TAFII -> H9.EZH2
H1.TAFII -> H1.MYC
H1.TAFII -> H1.NANOG
H1.TAFII -> H1.OCT4

null.vec
H1.H3K4me3 -> H9.RING1B
H1.H3K27me3 -> H9.RING1B
H1.H3K27me3 -> H1.expression
H1.H3K27me3 -> H1.mCGLevel
H1.H3K4me3 -> H1.H3K36me3
H1.NANOG -> H9.RING1B
H1.H3K27ac -> H1.H3K36me3
H1.H3K27me3 -> H1.H3K36me3
H1.H3K36me3 -> H1.expression
H1.H3K9ac -> H1.H3K36me3
H1.H3K4me3 -> H1.mCGLevel
H1.H3K27me3 -> H1.H3K9me3
H1.TAFII -> H1.mCGLevel
H9.EZH2 -> H9.RING1B
H1.H3K4me3 -> H1.expression
H1.H3K27ac -> H1.mCGLevel
H1.TAFII -> H1.OCT4
H1.H3K9ac -> H1.mCGLevel
H1.OCT4 -> H1.KLF4

H9.RING1B -> H1.mCGLevel
H1.H3K27me3 -> H1.OCT4
H1.H3K4me3 -> H1.KLF4
H1.TAFII -> H1.H3K9me3
H1.H3K4me3 -> H1.MYC
H1.TAFII -> H1.MYC
H1.TAFII -> H9.EZH2
H1.H3K27me3 -> H9.EZH2
H1.H3K4me3 -> H1.OCT4
H1.H3K9ac -> H9.EZH2

null.real
H1.H3K27ac -> H1.H3K9me3
H1.TAFII -> H1.H3K9me3
H9.EZH2 -- H1.TAFII
H1.MYC -> H1.KLF4
H1.OCT4 -> H1.KLF4
H1.H3K4me1 -> H1.H3K9me3
H1.H3K4me3 -> H1.KLF4
H9.EZH2 -> H1.H3K9me3
H1.H3K9ac -> H1.H3K9me3
H1.H3K27me3 -> H1.KLF4
H1.H3K27me3 -> H1.H3K9me3
H1.H3K4me3 -> H1.H3K36me3
H1.H3K9ac -> H1.expression
H1.H3K9ac -> H1.H3K36me3
H1.H3K27me3 -> H1.expression
H1.H3K4me3 -> H1.expression
H1.H3K27me3 -> H1.H3K36me3
H1.H3K27ac -> H1.H3K36me3
H1.H3K36me3 -> H1.expression
H1.NANOG -> H9.RING1B
H9.EZH2 -> H1.P300
H1.H3K27me3 -> H9.RING1B
H1.NANOG -> H1.P300
H9.EZH2 -> H9.RING1B
H1.H3K4me3 -> H1.MYC
H9.EZH2 -> H1.H3K27ac
H1.H3K4me3 -- H1.TAFII
H1.OCT4 -> H1.mCGLevel
H1.TAFII -> H1.H3K27ac
H1.OCT4 -> H1.MYC
H1.H3K4me3 -> H1.mCGLevel
H1.H3K9ac -- H1.TAFII

H1.NANOG -> H1.mCGLevel
H1.H3K27ac -> H1.mCGLevel
H9.EZH2 -- H1.H3K4me3
H1.H3K9ac -> H1.mCGLevel
H1.TAFII -> H1.MYC
H9.EZH2 -> H1.H3K36me3
H1.H3K27me3 -> H1.mCGLevel
H1.H3K4me1 -> H1.MYC
H9.EZH2 -> H1.mCGLevel
H9.EZH2 -> H1.MYC
H1.H3K27me3 -> H1.H3K27ac
H9.EZH2 -- H1.H3K9ac
H1.H3K27ac -> H1.MYC

sk.disc
H1.H3K27ac -> H1.expression
H9.EZH2 -> H9.RING1B
H1.SOX2 -> H9.EZH2
H9.EZH2 -> H1.expression
H9.EZH2 -> H1.H3K9me3
H1.H3K4me3 -> H1.H3K27me3
H1.H3K27ac -> H1.H3K36me3
H1.OCT4 -> H9.EZH2
H9.EZH2 -> H1.P300
H1.NANOG -- H1.TAFII
H1.H3K27me3 -> H1.mCGLevel
H1.NANOG -> H1.P300
H1.OCT4 -> H1.KLF4
H9.EZH2 -> H1.mCGLevel
H1.NANOG -> H1.mCGLevel
H1.TAFII -> H1.MYC
H1.mCGLevel -> H1.H3K9me3
H1.TAFII -> H9.RING1B
H9.EZH2 -> H1.H3K36me3
H1.TAFII -> H1.H3K4me3
H1.OCT4 -- H1.TAFII
H1.TAFII -> H1.H3K27me3
H1.H3K27ac -> H1.H3K27me3
H1.NANOG -- H1.SOX2

null.disc
H1.H3K27ac -> H1.H3K9me3
H1.mCGLevel -> H1.H3K9me3
H1.TAFII -> H1.H3K9me3

H1.OCT4 -> H1.KLF4
H1.mCGLevel -> H1.KLF4
H1.H3K4me3 -> H1.KLF4
H9.EZH2 -> H1.H3K9me3
H1.H3K4me1 -> H1.KLF4
H1.TAFII -> H9.RING1B
H1.H3K4me1 -- H1.TAFII
H1.H3K4me3 -- H1.TAFII
H1.H3K9ac -- H1.TAFII
H1.H3K27me3 -> H9.RING1B
H1.H3K27me3 -> H1.mCGLevel
H9.EZH2 -> H9.RING1B
H1.H3K27ac -- H1.H3K9ac
H1.H3K4me3 -- H1.H3K9ac
H1.OCT4 -> H1.MYC
H1.H3K27me3 -- H1.H3K4me3
H1.TAFII -> H1.MYC
H1.OCT4 -- H1.TAFII
H1.H3K27ac -- H1.TAFII
H1.H3K4me3 -> H9.RING1B

Dataset S3. The results generated by the literature co-citation analysis approach. All the relevant literature abstracts used in this work are listed with co-citied terms high-lighted.

(See cocitation_results.xls)

Dataset S4. The consensus hESC regulator networks derived from 6, 8, 12 or 14-bin tag profiles. See Table S3 for more details.

6 bins:
H1.H3K4me1 -- H1.H3K4me3
H1.H3K36me3 -- H1.expression
H1.KLF4 -- H1.OCT4
H1.H3K27ac -- H1.H3K4me1
H9.EZH2 -- H9.RING1B
H1.H3K27me3 -- H1.H3K9ac
H1.H3K4me3 -- H1.H3K9ac
H1.NANOG -- H1.SOX2
H1.H3K4me3 -- H1.OCT4
H1.H3K4me3 -- H1.expression
H1.H3K27me3 -- H9.EZH2
H1.NANOG -- H1.TAFII
H1.H3K4me3 -- H1.TAFII
H1.NANOG -- H1.P300
H1.MYC -- H1.TAFII

8 bins:
H1.H3K4me1 -- H1.H3K4me3
H1.H3K36me3 -- H1.expression
H1.KLF4 -- H1.OCT4
H1.NANOG -- H1.P300
H1.H3K27ac -- H1.H3K4me1
H1.H3K27me3 -- H9.EZH2
H9.EZH2 -- H9.RING1B
H1.H3K27me3 -- H1.H3K9ac
H1.H3K4me3 -- H1.H3K9ac
H1.NANOG -- H1.SOX2
H1.H3K4me3 -- H1.OCT4
H1.NANOG -- H1.TAFII
H1.H3K4me3 -- H1.MYC
H1.H3K4me3 -- H1.expression

12 bins:
H1.H3K27ac -- H1.H3K4me1
H1.H3K27me3 -- H1.H3K9ac
H1.H3K27me3 -- H9.EZH2
H1.H3K36me3 -- H1.expression
H1.H3K4me3 -- H1.H3K9ac
H1.H3K4me3 -- H1.OCT4

H1.H3K4me3 -- H1.TAFII
H1.H3K4me3 -- H1.expression
H1.H3K4me3 -- H1.mCGLevel
H1.KLF4 -- H1.OCT4
H1.NANOG -- H1.P300
H1.NANOG -- H1.SOX2
H9.EZH2 -- H9.RING1B

14 bins:
H1.H3K36me3 -- H1.expression
H1.KLF4 -- H1.OCT4
H1.H3K27ac -- H1.H3K4me1
H1.H3K27me3 -- H9.EZH2
H1.H3K4me3 -- H1.TAFII
H1.H3K27me3 -- H1.H3K9ac
H1.H3K4me3 -- H1.H3K9ac
H1.NANOG -- H1.SOX2
H1.H3K4me3 -- H1.OCT4
H1.NANOG -- H1.P300
H9.EZH2 -- H9.RING1B
H1.H3K4me3 -- H1.mCGLevel
H1.H3K4me3 -- H1.expression
H1.MYC -- H1.TAFII

Dataset S5. The constrained, hESC context-dependent motif interaction network (motif.vec.cons, shown in Figure S22a) and the general motif interaction network only based on the genomic sequence information (motif.vec.uncons, shown in Figure S11a), both learned from vectored representation of motif occurrences in the promoter regions.

motif.vec.cons:
EZH2_1_len11 -- H3K27me3_1_len11
H3K4me3_5_len10 -> TAFII_4_len11
H3K27me3_5_len10 -- H3K9ac_4_len11
H3K27me3_1_len11 -- H3K9ac_4_len11
H3K27me3_5_len10 -- H3K9ac_1_len11
H3K9ac_1_len11 -- H3K9ac_4_len11
EZH2_5_len11 -- H3K27me3_5_len10
MYC_10_len10 -- MYC_6_len10
H3K4me3_1_len11 -> TAFII_4_len11
H3K4me1_2_len11 -- H3K4me1_3_len11
TAFII_1_len11 -> TAFII_4_len11
H3K27me3_1_len11 -- H3K9ac_1_len11
EZH2_4_len11 -- EZH2_5_len11
EZH2_4_len11 -- H3K27me3_5_len10
H3K27me3_1_len11 -- H3K27me3_5_len10
EZH2_1_len11 -- H3K27me3_5_len10
TAFII_6_len11 -> TAFII_4_len11
KLF4_1_len11 -- KLF4_5_len11
H3K9ac_1_len11 -- H3K9ac_6_len11
EZH2_5_len11 -- H3K27me3_1_len11
H3K27me3_1_len11 -- H3K9ac_6_len11
EZH2_4_len11 -- H3K27me3_1_len11
NANOG_1_len10 -> P300_1_len11
TAFII_4_len11 -> NANOG_6_len11
H3K4me3_1_len11 -- H3K9ac_1_len11
H3K4me3_1_len11 -- OCT4_1_len11
H3K4me3_1_len11 -- H3K4me3_5_len10
KLF4_1_len11 -- OCT4_1_len11
H3K4me3_1_len11 -- MYC_10_len10
H3K4me1_2_len10 -- H3K4me1_3_len11
H3K4me3_5_len10 -- MYC_10_len10
KLF4_5_len11 -- OCT4_1_len11
H3K4me3_5_len10 -- H3K9ac_1_len11
H3K4me3_1_len11 -- MYC_6_len10
H3K27ac_2_len11 -- H3K4me1_3_len11
H3K4me3_1_len11 -- TAFII_6_len11
H3K4me1_2_len10 -- H3K4me3_1_len11

H3K4me3_1_len11 -- H3K9ac_4_len11
NANOG_6_len11 -> SOX2_5_len10
SOX2_5_len10 -> NANOG_1_len10
H3K4me3_5_len10 -- H3K9ac_4_len11
H3K4me1_3_len11 -- H3K4me3_1_len11
H3K4me3_1_len11 -- TAFII_1_len11
H3K27ac_2_len11 -- H3K4me1_2_len10
H3K27ac_2_len11 -- H3K4me1_2_len11
H3K4me3_5_len10 -- OCT4_1_len11
H3K4me3_5_len10 -- TAFII_6_len11
H3K4me1_2_len11 -- H3K4me3_1_len11
TAFII_1_len11 -- TAFII_6_len11
H3K4me1_2_len10 -- H3K4me1_2_len11
H3K4me1_3_len11 -- H3K4me3_5_len10
H3K4me1_2_len11 -- H3K4me3_5_len10
EZH2_5_len11 -- RING1B_5_len11

motif.vec.uncons:
H3K9ac_4_len11 -> TAFII_4_len11
MYC_10_len10 -> MYC_6_len10
TAFII_4_len11 -> OCT4_1_len11
MYC_10_len10 -> EZH2_5_len11
H3K4me3_5_len10 -> KLF4_5_len11
H3K4me3_1_len11 -> TAFII_4_len11
H3K4me3_1_len11 -> H3K9ac_4_len11
H3K27me3_5_len10 -> H3K9ac_4_len11
H3K4me3_1_len11 -> KLF4_5_len11
H3K9ac_4_len11 -> KLF4_5_len11
H3K27me3_5_len10 -> MYC_6_len10
H3K27me3_5_len10 -> EZH2_4_len11
EZH2_5_len11 -> EZH2_4_len11
H3K4me3_5_len10 -> TAFII_4_len11
H3K27me3_5_len10 -> TAFII_4_len11
H3K4me3_5_len10 -> OCT4_1_len11
H3K9ac_1_len11 -> H3K9ac_4_len11
H3K27me3_5_len10 -> EZH2_5_len11
H3K9ac_4_len11 -> OCT4_1_len11
TAFII_1_len11 -> MYC_6_len10
MYC_10_len10 -> EZH2_4_len11
H3K27me3_1_len11 -> MYC_6_len10
H3K9ac_4_len11 -> NANOG_6_len11
TAFII_1_len11 -> EZH2_5_len11
H3K9ac_4_len11 -> EZH2_5_len11
H3K4me3_5_len10 -> H3K9ac_4_len11

H3K4me1_2_len10 -- H3K4me1_3_len11
H3K4me1_3_len11 -- KLF4_1_len11
H3K4me1_2_len11 -- H3K4me1_3_len11
H3K4me1_2_len10 -- H3K4me1_2_len11
H3K27ac_2_len11 -> EZH2_5_len11
H3K4me1_2_len11 -- KLF4_1_len11
H3K4me1_2_len10 -- KLF4_1_len11
H3K4me1_2_len10 -> MYC_10_len10
H3K9ac_4_len11 -> RING1B_1_len11
H3K4me1_3_len11 -> MYC_10_len10
H3K4me1_3_len11 -- H3K4me3_1_len11
TAFII_4_len11 -> KLF4_5_len11
H3K9ac_1_len11 -> MYC_10_len10
TAFII_6_len11 -> TAFII_4_len11
H3K4me1_2_len11 -- H3K4me3_1_len11
RING1B_5_len11 -> MYC_4_len10
TAFII_4_len11 -> MYC_6_len10
NANOG_1_len10 -- SOX2_5_len10
SOX2_5_len10 -> NANOG_6_len11
P300_1_len11 -> SOX2_5_len10
P300_1_len11 -> NANOG_1_len10
H3K4me1_2_len10 -- H3K4me3_1_len11
TAFII_6_len11 -> H3K9ac_4_len11
TAFII_1_len11 -> H3K9ac_4_len11
H3K27me3_1_len11 -> MYC_10_len10
H3K9ac_4_len11 -> P300_1_len11

References

1 Koller D, Friedman N. Probabilistic Graphical Models - Principles and Techniques: MIT Press 2009.

2 Bach FR, Jordan MI. Learning Graphical Models with Mercer Kernels. In: Becker S, Thrun S, Obermayer K. eds. Advances in Neural Information Processing Systems. Vancouver, Canada: MIT Press 2002:1009-1016.

3 Bach FR, Jordan MI. Kernel Independent Component Analysis. *Journal of Machine Learning Research* 2002; **3**:1-48.

4 Rubner Y, Tomasi C, Guibas LJ. The Earth Mover's Distance as a metric for image retrieval. *International Journal of Computer Vision* 2000; **40**:99-121.

5 Ling HB, Soatto S. Proximity distribution kernels for geometric context in category recognition. *2007 IEEE 11th International Conference on Computer Vision, Vols 1-6* 2007:245-252.

6 Heckerman D, Chickering DM, Meek C, Rounthwaite R, Kadie C. Dependency Networks for Inference, Collaborative Filtering,and Data Visualization. *Journal of Machine Learning Research* 2000; **1**:49-75.

7 van Steensel B, Braunschweig U, Filion GJ, Chen M, van Bemmel JG, Ideker T. Bayesian network analysis of targeting interactions in chromatin. *Genome Res* 2010; **20**:190-200.

8 Yu H, Zhu SS, Zhou B, Xue HL, Han JDJ. Inferring causal relationships among different histone modifications and gene expression. *Genome Res* 2008; **18**:1314-1324.

9 Arthur D, Vassilvitskii S. k-means plus plus : The Advantages of Careful Seeding. *Proceedings of the Eighteenth Annual Acm-Siam Symposium on Discrete Algorithms* 2007:1027-1035.

10 Hartigan JA. Clustering algorithms. New York,: Wiley 1975.

11 Lister R, Pelizzola M, Dowen RH *et al.* Human DNA methylomes at base resolution show widespread epigenomic differences. *Nature* 2009; **462**:315-322.

12 Ku M, Koche RP, Rheinbay E *et al.* Genomewide analysis of PRC1 and PRC2 occupancy identifies two classes of bivalent domains. *PLoS Genet* 2008; **4**:e1000242.

13 Jiang H, Wong WH. Statistical inferences for isoform expression in RNA-Seq. *Bioinformatics* 2009; **25**:1026-1032.

14 Mortazavi A, Williams BA, McCue K, Schaeffer L, Wold B. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat Methods* 2008; **5**:621-628.

15 Heckerman D, Geiger D, Chickering DM. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Mach Learn* 1995; **20**:197-243.

16 Steck H. Learning the Bayesian Network Structure: Dirichlet Prior vs Data. In: McAllester DA, Myllymaki P. eds. Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence. Helsinki, Finland: AUAI Press 2008:511-518.

17 Meek C. Causal inference and causal explanation with background knowledge. *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence* 1995:403-410.

18 Chickering DM. A Transformational Characterization of Equivalent Bayesian Network Structures. *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence* 1995:87-98.

19 Zang C, Schones DE, Zeng C, Cui K, Zhao K, Peng W. A clustering approach for identification of enriched domains from histone modification ChIP-Seq data. *Bioinformatics* 2009; **25**:1952-1958.

20 Zhang Y, Liu T, Meyer CA *et al.* Model-based analysis of ChIP-Seq (MACS). *Genome Biol* 2008; **9**:R137.

21 Smith AD, Sumazin P, Zhang MQ. Identifying tissue-selective transcription factor binding sites in vertebrate promoters. *Proc Natl Acad Sci U S A* 2005; **102**:1560-1565.

22 Smith AD, Sumazin P, Das D, Zhang MQ. Mining ChIP-chip data for transcription factor and cofactor binding sites. *Bioinformatics* 2005; **21 Suppl 1**:i403-412.
23 Smith AD, Sumazin P, Xuan Z, Zhang MQ. DNA motifs in human and mouse proximal promoters predict tissue-specific expression. *Proc Natl Acad Sci U S A* 2006; **103**:6275-6280.
24 Schones DE, Smith AD, Zhang MQ. Statistical significance of cis-regulatory modules. *BMC Bioinformatics* 2007; **8**:19.
25 Geiger D, Heckerman D. Learning Gaussian Networks. *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence* 1995:235-243.
26 Spirtes P, Glymour C, Scheines R. Causation, Prediction, and Search, 2nd ed. *MIT Press* 2000.
27 Friedman N, Linial M, Nachman I, Pe'er D. Using Bayesian networks to analyze expression data. *J Comput Biol* 2000; **7**:601-620.
28 Sachs K, Perez O, Pe'er D, Lauffenburger DA, Nolan GP. Causal protein-signaling networks derived from multiparameter single-cell data. *Science* 2005; **308**:523-529.
29 Lloyd SP. Least-Squares Quantization in Pcm. *IEEE Transactions on Information Theory* 1982; **28**:129-137.
30 Frey BJ, Dueck D. Clustering by passing messages between data points. *Science* 2007; **315**:972-976.
31 Yi BK, Jagadish HV, Faloutsos C. Efficient retrieval of similar time sequences under time warping. *14th International Conference on Data Engineering, Proceedings* 1998:201-208.
32 Heintzman ND, Stuart RK, Hon G *et al.* Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome. *Nat Genet* 2007; **39**:311-318.
33 Heintzman ND, Hon GC, Hawkins RD *et al.* Histone modifications at human enhancers reflect global cell-type-specific gene expression. *Nature* 2009; **459**:108-112.
34 Boyer LA, Lee TI, Cole MF *et al.* Core transcriptional regulatory circuitry in human embryonic stem cells. *Cell* 2005; **122**:947-956.
35 Zhong X, Jin Y. Critical roles of coactivator p300 in mouse embryonic stem cell differentiation and Nanog expression. *J Biol Chem* 2009; **284**:9168-9175.
36 Fong H, Hohenstein KA, Donovan PJ. Regulation of self-renewal and pluripotency by Sox2 in human embryonic stem cells. *Stem Cells* 2008; **26**:1931-1938.
37 Kim J, Chu J, Shen X, Wang J, Orkin SH. An extended transcriptional network for pluripotency of embryonic stem cells. *Cell* 2008; **132**:1049-1061.
38 Hawkins RD, Hon GC, Lee LK *et al.* Distinct epigenomic landscapes of pluripotent and lineage-committed human cells. *Cell Stem Cell* 2010; **6**:479-491.
39 Rahl PB, Lin CY, Seila AC *et al.* c-Myc Regulates Transcriptional Pause Release. *Cell* 2010; **141**:432-445.
40 Guenther MG, Levine SS, Boyer LA, Jaenisch R, Young RA. A chromatin landmark and transcription initiation at most promoters in human cells. *Cell* 2007; **130**:77-88.
41 Hon G, Wang W, Ren B. Discovery and Annotation of Functional Chromatin Signatures in the Human Genome. *Plos Computational Biology* 2009; **5**:e1000566.
42 Ang YS, Tsai SY, Lee DF *et al.* Wdr5 mediates self-renewal and reprogramming via the embryonic stem cell core transcriptional network. *Cell* 2011; **145**:183-197.
43 Vakoc CR, Sachdeva MM, Wang H, Blobel GA. Profile of histone lysine methylation across transcribed mammalian chromatin. *Mol Cell Biol* 2006; **26**:9185-9195.
44 Kim T, Buratowski S. Dimethylation of H3K4 by Set1 recruits the Set3 histone deacetylase complex to 5' transcribed regions. *Cell* 2009; **137**:259-272.

45 Ong CT, Corces VG. Enhancer function: new insights into the regulation of tissue-specific gene expression. *Nature Reviews Genetics* 2011; **12**:283-293.
46 Langmead B, Trapnell C, Pop M, Salzberg SL. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol* 2009; **10**:R25.
47 Li R, Yu C, Li Y *et al.* SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics* 2009; **25**:1966-1967.
48 Xiao S, Xie D, Cao X *et al.* Comparative epigenomic annotation of regulatory DNA. *Cell* 2012; **149**:1381-1392.
49 van den Berg DL, Snoek T, Mullin NP *et al.* An Oct4-centered protein interaction network in embryonic stem cells. *Cell Stem Cell* 2010; **6**:369-381.
50 Baltus GA, Kowalski MP, Zhai H *et al.* Acetylation of sox2 induces its nuclear export in embryonic stem cells. *Stem Cells* 2009; **27**:2175-2184.
51 Levine M. Transcriptional enhancers in animal development and evolution. *Curr Biol* 2010; **20**:R754-763.
52 Yu H, Tardivo L, Tam S *et al.* Next-generation sequencing to generate interactome datasets. *Nat Methods* 2011; **8**:478-480.