Corrections

NEUROBIOLOGY


The authors note that Figure 3 appeared incorrectly. The corrected figure and its legend appear below. This error does not affect the conclusions of the article.

![Image of corrected figure](https://www.pnas.org/doi/10.1073/pnas.1311228110)

**Fig. 3.** Upper and lower attachments of the tip link. (A and B) Freeze-etch images of tip-link upper insertions in guinea pig cochlea (A) and (left to right) two from guinea pig cochlea, two from bullfrog sacculus, and two from guinea pig utriculus (B). Each example shows pronounced branching. (C and D) Freeze-etch images of the tip-link lower insertion in stereocilia from bullfrog sacculus (C) and guinea pig utriculus (D); multiple strands (arrows) arise from the stereociliary tip. (E) Freeze-fracture image of stereociliary tips from bullfrog sacculus; indentations at tips are indicated by arrows. (Scale bars: A = 100 nm, B = 25 nm; C–E = 100 nm.)

www.pnas.org/doi/10.1073/pnas.1311228110
BIOPHYSICS AND COMPUTATIONAL BIOLOGY, STATISTICS
Correction for “Differential principal component analysis of ChIP-seq,” by Hongkai Ji, Xia Li, Qian-fei Wang, and Yang Ning, which appeared in issue 17, April 23, 2013, of Proc Natl Acad Sci USA (110:6789–6794; first published April 8, 2013; 10.1073/pnas.1204398110).

The authors note the following statement should be added to the Acknowledgments: “Q.-F.W. is supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDA01010305).”

www.pnas.org/cgi/doi/10.1073/pnas.1311614110
We propose differential principal component analysis (dPCA) for analyzing multiple ChIP-sequencing datasets to identify differential protein–DNA interactions between two biological conditions. dPCA integrates unsupervised pattern discovery, dimension reduction, and statistical inference into a single framework. It uses a small number of principal components to summarize concisely the major multiprotein synergistic differential patterns between the two conditions. For each pattern, it detects and prioritizes differential genomic loci by comparing the between-condition differences with the within-condition variation among replicate samples. dPCA provides a unique tool for efficiently analyzing large amounts of ChIP-sequencing data to study dynamic changes of gene regulation across different biological conditions. We demonstrate this approach through analyses of differential chromatin patterns at transcription factor binding sites and promoters as well as allele-specific protein–DNA interactions.

ChIP coupled with high-throughput sequencing (seq) is increasingly used for studying transcription factor (TF) binding sites and histone modifications (HMs) (1–3). A fundamental but unsolved problem in ChIP-seq data analysis is to compare quantitative binding signals between two biological conditions with respect to multiple proteins. This problem is often raised when researchers study changes of gene regulatory programs between different conditions (e.g., normal and cancer cells, different developmental time points). Fig. 1 shows a representative data structure. For each of the two conditions, data for multiple TFs and/or HMs are available. Each protein may have multiple replicate samples carrying information about the biological or technical variation. Given these data, one often asks three questions: (i) What are the major patterns of protein–DNA interaction (PDI) differences between the two conditions? (ii) How do I detect and prioritize differential genomic loci for follow-up studies? (iii) How do I evaluate statistical significance of the observed differences, given the background variation among replicate samples?

These fundamental questions cannot be answered by existing ChIP-seq data analysis tools. Most peak-calling algorithms are developed for finding PDI loci in one cell type (4, 5) (see also SI Appendix, Text S1). They do not characterize quantitative differences between two cell types. Although one can compare two cell types based on the binary peak calls, this comparison is qualitative and cannot replace a quantitative comparison of the continuous binding signals. For instance, a peak called present in one condition and absent in the other does not provide any information about the strength of binding. A few methods that incorporate some quantitative elements (e.g., ChIP-peak intensity) are available (6–12). Analyzing ChIP-seq quantitatively allows one to distinguish between continuous binding signals. For instance, a peak called present in one condition and absent in the other does not provide any information about the strength of binding. A few methods that incorporate some quantitative elements (e.g., ChIP-peak intensity) are available (6–12).

For each locus, statistical significance is evaluated by comparing the between-condition differences with the background variation among replicate samples. dPCA takes into account both within-condition variation among replicates and between-condition differences, which are generated by adding independent Gaussian random noise to true binding levels \( \mu_{gimk} \). The variance of \( \tilde{\sigma}_{gimk}^2 \), \( \sigma_{gimk}^2 \), characterizes the variability among replicates and is unknown. Taking an average over replicates gives \( \bar{\sigma}_{gimk} = \frac{1}{m} \sum_k \sigma_{gimk} \) and \( \tilde{\sigma}_{gimk}^2 = \frac{1}{m} \sum_k \sigma_{gimk}^2 \) (13). This removes the dependency on the number of replicates and leads to a more robust ranking of differential binding signals.

By applying dPCA to ChIP-seq data from both simulations and real data, we find that this approach has two major drawbacks. First, analyzing each dataset separately ignores the correlation among proteins that may provide insight into multiprotein synergy. Second, if there are \( M \) datasets, this approach will produce \( M \) differential loci lists and \( 3^M \) combinatorial patterns (because each locus has three possible states in each dataset: up, down, and no change). As \( M \) grows, the results will become difficult to report, interpret, and use. For example, if one wants to choose some differential loci to follow up experimentally, which of the \( M \) lists or \( 3^M \) patterns should be followed up first, given the finite resource?

To address these issues, dPCA integrates unsupervised pattern discovery, dimension reduction, and statistical tests into a single framework. It first summarizes main patterns of differences between the two biological conditions using a small number of differential principal components (dPCs). Each dPC represents a covariation pattern of quantitative signals among multiple proteins. dPCs can simplify description of the data. The analysis then identifies differential genomic loci for each major dPC and prioritizes these loci based on their magnitude of differences. For each locus, statistical significance is evaluated by comparing the between-condition differences with the background variation among the replicate samples. We will demonstrate dPCA using both simulations and real data. dPCA is implemented in ANSI C and is freely available at www.biostat.jhsph.edu/dpca.

Results

dPCA. Consider two biological conditions (\( i = 1, 2 \)), each with \( M \) datasets. In condition \( i \), dataset \( m \) has \( K_m \) replicates. There are \( G \) genomic loci. Typically \( G \gg M \) (Fig. 1). dPCA takes coordinates of these loci and aligned ChIP-seq reads as input. After preprocessing, normalization, and log2 transform (SI Appendix, Text S1), PDI intensity for locus \( g \), condition \( i \), dataset \( m \), and replicate \( k \) will be summarized into one value: \( x_{gimk} \). We assume that \( x_{gimk} \)s are generated by adding independent Gaussian noises \( \tilde{\epsilon}_{gimk} \) to true binding levels \( \mu_{gim} \). The variance of \( \tilde{\epsilon}_{gimk} \) is \( \tilde{\sigma}_{gimk}^2 \), which characterizes the variability among replicates and is unknown. Taking an average over replicates gives \( \bar{\sigma}_{gim} = \frac{1}{m} \sum_k \sigma_{gimk} \) and \( \tilde{\sigma}_{gim}^2 = \frac{1}{m} \sum_k \sigma_{gimk}^2 \) (13).

The authors declare no conflict of interest.

This article is a PNAS Direct Submission. H.Z. is a guest editor invited by the Editorial Board. Freely available online through the PNAS open access option.

1To whom correspondence should be addressed. E-mail: hj@jhsph.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1204398110/-/DCSupplemental.
(1) Summarize data into two matrices

<table>
<thead>
<tr>
<th>Cell Type 1</th>
<th>Cell Type 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset:</td>
<td>Dataset:</td>
</tr>
<tr>
<td><a href="1000">(HMK)</a></td>
<td><a href="1000">(HMK)</a></td>
</tr>
<tr>
<td><a href="1000">(HMY)</a></td>
<td><a href="1000">(HMY)</a></td>
</tr>
<tr>
<td><a href="1000">(HMR)</a></td>
<td><a href="1000">(HMR)</a></td>
</tr>
<tr>
<td><a href="1000">(M1M)</a></td>
<td><a href="1000">(M1M)</a></td>
</tr>
<tr>
<td><a href="1000">(M2M)</a></td>
<td><a href="1000">(M2M)</a></td>
</tr>
<tr>
<td><a href="1000">(M3M)</a></td>
<td><a href="1000">(M3M)</a></td>
</tr>
<tr>
<td>[Replicate:]</td>
<td>[Replicate:]</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

(2) Take average across replicates

\[ \bar{X}_1 = (\bar{x}_{1m})_{G \times M} = \frac{1}{n} \sum_{i=1}^{n} x_{im} \]

(3) Compute observed difference

\[ D_{\text{GM}} = \text{True Difference} \Lambda + \text{Noise} E \]

(4) Decompose observed difference

\[ \Lambda = B V^T \]

(5) Decompose unobserved true difference

\[ \delta^i = v_i^T \delta \]

(6) Use the first R principal components (red and green) to characterize major differences

\[ d_{GM} = x_{GM} - x_{GM}. \]

The observed difference between the two conditions for locus g and dataset m is \( d_{GM} \).

Organize \( d_{GM} \) into a matrix \( D = (d_{GM})_{G \times M} \). Each row of \( D \) corresponds to a locus, and each column corresponds to a dataset. We consider applications in which, within each column, \( d_{GM} \) is both positive and negative and fluctuate around zero (SI Appendix, Text S1). We decompose \( D \) into two matrices: \( D = \Delta + E \), where \( \Delta = (\delta_{GM})_{G \times M} \) represents the unobserved true differences between the two conditions and \( E = (\epsilon_{GM})_{G \times M} \) corresponds to random sampling noise. Here, \( \delta_{GM} \sim \mathcal{N}(0, \sigma^2(1/K_{1m} + 1/K_{2m})) \). The \( \epsilon_{GM} \) are independent and reflect replicate variation. The gth row of matrices \( \Delta, \Lambda, \text{ and } E \), denoted by \( \delta^g, \delta^g, \text{ and } \epsilon^g \), comprises the observed difference, unobserved true difference, and sampling noise at locus g, respectively. Here, \( T \) means vector or matrix transpose.

Our primary interest is the unobserved truth \( \Delta \). Intuitively, dPCA attempts to characterize \( \Delta \) by a few principal components (PCs). This is different from the conventional principal component analysis (PCA) that studies PCs of the observed data matrix \( D \). Distinguishing \( \Delta \) from \( D \) is important because it allows one to assess how much variation in the observed \( D \) is due to the underlying truth (\( \Delta \)) as opposed to the variation among replicate samples (\( E \)). Subsequently, one will be able to infer whether the estimated PCs can accurately match the truth, assess the statistical significance of the observed differences, and more efficiently reduce the data dimension. These are functions not provided by PCA.

dPCA is based on assuming that there exist \( M \) orthogonal differential patterns \( v_1, \ldots, v_M \) such that the true difference \( \delta \) at each locus can be represented by a linear combination:

\[ \delta_g = \sum_{j=1}^{M} \beta_{gj} v_j = \beta_g v \]

Each \( v_j \) is a \( M \times 1 \) vector with unitary length, representing a covariance pattern of binding intensities among multiple ChIP-seq datasets (Fig. 1). \( V_{\text{full}} = (v_1, \ldots, v_M) \) is an orthogonal matrix (i.e., \( V^T V = I \)). We treat \( V \) as fixed but unknown parameters. Given \( V \), \( \delta_g \) are assumed to be random and independently generated as follows. First, 0/1 valued binary indicators \( b_{gj} \) are independently drawn from Bernoulli distributions with success probability \( Pr(b_{gj} = 1) = \pi_j \). Second, real-valued coefficients \( w_j \) are drawn independently from some unknown distributions \( H_i(w; \mu_i, \tau_i^2) \) with zero mean and unknown variance \( \tau_i^2 \). The products \( \beta_{gj} \equiv b_{gj} \times w_j \) are then used as the coefficients to combine \( v_j \) to generate \( \delta_g \). In Eq. 1, \( V \) is common to all loci, but \( \beta_{gj} \) are locus-specific. This model implies that each locus can be differential with respect to some patterns (when \( \beta_{gj} = 0 \)) but nondifferential for the others (\( \beta_{gj} = 0 \)). For each pattern \( v_j \), the data consist of a mixture of differential and nondifferential loci, and \( \pi_j \) is the prior probability for a locus to be differential.

Group coefficients \( \beta_{gj} \) into a matrix \( B = (\beta_{gj})_{G \times M} \). The jth row of the matrix. It contains all coefficients \( \beta_{gj} \) for locus g. The jth column of B contains all coefficients for pattern \( v_j \). In matrix form, Eq. 1 is equivalent to \( \Delta = BV^T \) (Fig. 1). Based on our model, \( \beta_{gj} \) in column j of B have a mean \( E(\beta_{gj}) = 0 \) and variance \( \text{Var}(\beta_{gj}) = \pi_j \tau_j^2 \); hence, \( \lambda_j \) characterizes the variation in \( \Delta \) contributed by pattern \( v_j \). We assume that \( \beta_{gj} \) are unequal and, without loss of generality, arranged in descending order \( \lambda_1 > \ldots > \lambda_M \geq 0 \). Under this arrangement, \( \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_M) \) is a diagonal matrix and \( \lambda_j \) are its diagonal elements. Thus, \( \lambda_j \) are the unique eigenvalues of \( \text{Var}(\beta_{gj}) \) due to the uniqueness of eigenvalue decomposition, and \( v_j \) are the unique eigenvectors up to a multiplier of \( \pm 1 \), which will not affect the two-sided hypothesis tests below. Each \( \lambda_j \) essentially corresponds to a PC of \( \text{Var}(\beta_{gj}) \) and is called a dPC. In real data, the first few dPCs often explain the main variation in \( \Delta \) (i.e., one can find an integer \( R \ll M \) such that \( \sum_{j=1}^{R} \lambda_j / \sum_{j=1}^{M} \lambda_j \) is big). Consequently, one can use the first R dPCs instead of all M patterns to summarize the major changes between conditions (i.e., \( \Delta \approx B G \times R, X_{GM} \)), where \( B \) and \( V \) contain the first R columns of \( B \) and \( V \), respectively. Reducing the dimension from \( M \) to \( R \) can greatly reduce the complexity of data interpretation and make follow-up studies more manageable.

In the model above, \( V, B, \sigma^2, \pi_j \), and \( H_i(0; 0, \tau_i^2) \) are all unknown. Our primary interest is to \( V \) and \( B \). dPCA has three goals: (i) find the major differential patterns \( V \); (ii) for each locus g and pattern \( v_j \), estimate \( \beta_{gj} \) by projecting data to the estimated \( v_j \) (i.e., \( d_{GM} = y_{GM}^T \delta + e_{GM} \)). Hence, we infer whether the locus is differential or not, i.e., test \( H_i; \beta_{gj} = 0 \) vs. \( H_i; \beta_{gj} \neq 0 \); and (iii) for each pattern \( v_j \), rank genomic loci based on the magnitude of difference \( \beta_{gj} \) for follow-up studies. We developed a computationally efficient algorithm to achieve these goals (Methods). For the examples below, the algorithm only takes 1–2 min on a laptop computer with a 2.2 GHz central processing unit (CPU) and 4 GB of random access memory after data preprocessing, which takes a much longer time.

To determine which dPCs to report, we project data to each dPC and define a signal-to-noise ratio (SNR) measure \( \text{SNR} = \text{Var}(v_j^T d_{GM})/\text{Var}(v_j^T e_{GM}) \). We estimate \( \text{SNR} \) and report leading dPCs for which \( \text{SNR} > 0.5 \). This is based on observing that the dPC estimates and statistical inference on \( \beta_{gj} \) are not reliable when the SNR is small (examples I–III).

The basic model above analyzes differences without considering the total amount of absolute binding at each locus. Users often want to analyze differences more specifically at locations where there are significant binding activities that might be easier to interpret or to study experimentally. We provide multiple options to do so. For instance, one can filter out loci not bound in any dataset before dPCA. This and other more sophisticated options are discussed in detail in SI Appendix, Text S1.

dPCA is different from factor analysis (SI Appendix, Text S1). Unlike methods requiring random initialization or ad hoc choice of parameters (e.g., K-means clustering), dPCA is a deterministic algorithm and patterns discovered by dPCA are reproducible from one investigator to another.
Example I: Analysis of Differential Chromatin Patterns at TF Binding Sites. We first demonstrate dPCA using 18 ENCODE (3) chromatin datasets consisting of 70 ChIP-seq, DNase-seq, and Formaldehyde-Assisted Isolation of Regulatory Elements (FAIRE)-seq samples from two cell lines: K562 and human umbilical vein endothelial cell (HUVEC) (SI Appendix, Table S1). Each dataset had one to three replicates in each cell line. We mapped the MYC (E-box) motif to the human genome and obtained 138,325 MYC motif sites. After excluding sites not associated with significant chromatin signal(s) in any dataset, dPCA was applied to explore differential chromatin patterns at the remaining 58,997 motif sites (Fig. 2 and SI Appendix, Text S1 and Fig. S2).

We begin with asking whether the differential patterns discovered by dPCA are biologically meaningful. The top two dPCs passed the cutoff of \( \text{SNR} > 5 \). They explained 58.7% and 17.2% of the variance in \( \delta \lambda \) (5,000 top sites) were distinctly different (Fig. 2A and B). In dPC1, differences between the two cell types are primarily driven by H3K4me1, H3K4me2, H3K4me3, H3K9ac, and H3K27ac. These HMs are known marks for active transcription or enhancer activities. dPC2 mainly captures the difference in H3K27me3, which is a mark for gene repression. Thus, without using prior knowledge, these two dPCs automatically summarized 18 datasets into two biologically meaningful modules corresponding to gene activation and repression, respectively.

We then asked whether dPCA provides a meaningful way to rank differential patterns for follow-up studies. Using independent ENCODE MYC ChIP-seq data, we computed \( \log_2 \) fold changes (\( \log_2 FC \)) of MYC ChIP-seq signals between the two cell lines (SI Appendix, Text S1). Interestingly, although our dPCA analysis did not involve any MYC ChIP-seq data, the coefficients \( \beta_j \) for dPC1 strongly correlated with the differential MYC ChIP-seq signals (Fig. 2D; Pearson’s correlation, \( r = 0.65 \)). We further defined 6,433 motif sites bound by MYC in at least one cell type and with MYC ChIP-seq \( \log_2 FC > 1.5 \) as true differential MYC binding sites (SI Appendix, Text S1). Fig. 2E shows that a significant fraction of the top motif sites ranked by dPC1 (e.g., 2,995 of 5,000 top sites) were indeed differentially bound by MYC. Importantly, compared with motif site rankings based on each individual dataset (using \( \delta \lambda \) as the ranking criterion), the dPC1 ranking predicted differential MYC binding better (Fig. 2E and SI Appendix, Text S1 and Fig. S2D). Moreover, if one were to use the best single dataset-based ranking to choose differential loci for follow-up study, one would have to determine which of the 18 datasets is the best. If there were no prior knowledge or independent benchmark data, such as MYC ChIP-seq, this would be difficult. The unsupervised dPCA produced the best ranking without using any prior knowledge. It is able to integrate information automatically from multiple datasets and to prevent one from being overwhelmed by having too many datasets. Unlike dPC1, dPC2 only had a weak negative correlation with differential MYC binding (\( r = -0.06 \); Fig. 2D and E). This weak correlation mainly reflects the nature of the H3K27me3 data and could have many possible explanations (SI Appendix, Text S1). Jointly, dPC1 and dPC2 were able to explain the differential MYC binding slightly better (SI Appendix, Text S1 and Fig. S2B and E).

At the 5% false discovery rate (FDR) level, dPCA reported 34,034 (57.7% of 58,997 and 24.6% of 138,325) and 28,379 (48.1% of 58,997 and 20.5% of 138,325) differential motif sites for dPC1 and dPC2, respectively, with 16,906 common sites (Fig. 2C). This amounts to a total of 45,507 (77.1% of 58,997 and 32.9% of 138,325) differential sites. Without knowing the truth, it is difficult to evaluate how accurate the dPC and FDR estimates are. To shed light on the performance of these estimates, we performed simulations by retaining the main characteristics of real data, which may deviate from the assumptions made by dPCA (e.g., normality, common \( \sigma^2 \) for all loci). Simulations were performed in different global SNR settings (Fig. 3 and SI Appendix, Text S3), with details described in SI Appendix, Text S1. Fig. 3 provides a representative example to illustrate the results. Fig. 3A shows the estimated SNR for each dPC. Fig. 3B shows the accuracy of the \( \psi_j \) estimates. The accuracy was measured by the cosine distance \( \delta(v_j, v_i) = 1 - \cos(v_j, v_i) \). A small \( d \) means accurate. The vertical bars show the variability of \( d \), measured by its SD across 10 independent simulations. Fig. 3C shows the error of \( \lambda_j \) estimates (i.e., \( \lambda_j - \lambda_j \)). Fig. 3D shows the percentage of variance explained by the top dPCs. Fig. 3E–G compares the true FDR with the estimated FDR for the first three dPCs, respectively. These results show that the accuracy of dPC estimates decreases with decreasing \( \text{SNR} \) (Fig. 3A and B). For dPCs with \( \text{SNR} > 5 \), the estimated \( \psi_j \) matched the true \( \psi_j \) well and the estimated FDR provided reasonable estimates for the true FDR for testing \( \beta_j = 0 \) vs. \( \beta_j \neq 0 \) even if the data were projected to \( \psi_j \) instead of \( \psi_j \) (Fig. 3B and E). The performance deteriorated as \( \text{SNR} \) decreased (Fig. 3B and F). For dPCs with \( \text{SNR} < 5 \), the estimates were off the mark (Fig. 3B and G). For dPCs with a small \( \text{SNR} \), \( \psi_j \) also had high variability, as evidenced by the wide error bars in Fig. 3B and additional simulations in SI Appendix, Text S1 and Fig. S4 A–C, which show that if two laboratories independently generate similar data and run dPCA, they may not discover the same patterns, causing a reproducibility issue. Intuitively, when \( \text{SNR} \) is small, the geometric direction represented by \( \psi_j \) in the \( \mathbb{R}^n \) space can be easily rotated by noise. When \( \psi_j \) is biased, it is not reliable to draw conclusions about \( \beta_j = \psi_j \beta_j \) by projecting data to \( \psi_j \), because \( \psi_j \) and \( \psi_j \) represent different geometric directions. We observed similar phenomena in all simulations (SI Appendix, Fig. S3). Therefore, although all nonzero elements in \( B \) are assumed to be true differences not explained by cross-sample variation, we only report dPCs with \( \text{SNR} > 5 \) (dPC1 and dPC2 in this example), because the differential patterns with a smaller \( \text{SNR} \) cannot be accurately and...
reproducibly discovered. In the real data, dPC2 has $\text{SNR} > 5$. Based on the simulations, it is very likely to be a true differential pattern despite its weak correlation with differential MYC binding.

In our data, the top two dPCs had patterns similar to the top two PCs in PCA (Fig. 2A and SI Appendix, Fig. S2G). However, the top two PCs in PCA only explained 57% of the variance in $\mathbf{D}$, whereas the top two dPCs explained 76% of the variance in $\mathbf{D}$. To explain the $\geq 76\%$ of the variance in $\mathbf{D}$, PCA needs six PCs. To reduce dimension, the conventional PCA often chooses the number of PCs based on the percentage of variance explained. Using this criterion, dPCA is more efficient for dimension reduction. This is confirmed by simulations showing that the eigenvalues in PCA tend to be bigger than those in dPCA (Fig. 3C), which results in a smaller percentage of variance explained by the top PCs (Fig. 3D); an intuitive explanation is provided in SI Appendix, Text S1). Unlike PCA, dPCA also provides $\text{SNR}$ to help one choose which dPCs to report based on judging whether the dPC and FDR estimates are close to the truth and whether dPCs are reproducible in future studies.

Our analysis suggests that one can combine motif analysis with surrogate experiments, such as HM ChIP-seq to infer dynamic changes of TF binding. Analyses of several other TFs, cell types, and data combinations confirmed this observation (SI Appendix, Table S1 and Fig. S2 L–N). Performing ChIP-seq experiments for all TFs is currently not feasible due to a lack of antibodies and the high cost. However, good antibodies for many HMs are available, and among the 1,400 human TFs, $\sim 500$ have known DNA binding motifs. Therefore, dPCA analysis of multiple surrogate datasets (e.g., HM ChIP-seq) provides a solution to unsupervised characterization of gene regulation dynamics, and it allows one to infer differential binding of many TFs simultaneously using the same set of experiments. Unlike several recent studies that use surrogates to predict TF binding in one condition (19, 20), dPCA allows one to predict dynamic changes of TF binding across conditions.

**Example II: Analysis of Differential Promoters.** We also analyzed 24,376 human promoters using the same 18 datasets in K562 and Huvec lines (Fig. 4 and SI Appendix, Text S1, Table S1, and Fig. S5). Applying dPCA to the 22,368 promoters bound in at least one dataset, two dPCs passed the cutoff of $\text{SNR} > 5$. They were similar to the ones found in the MYC analysis, except that H3K4me1 played a weaker role in dPC1 in the promoter analysis (Figs. 2A and 4A). This is consistent with the knowledge that H3K4me1 preferentially marks enhancers rather than promoters (21). At the 5% FDR level, 16,990 (76.0% of 22,368 and 69.7% of 24,376) and 13,735 (61.4% of 22,368 and 56.4% of 24,376) differential promoters (common = 10,818, total = 19,907) were found for dPC1 and dPC2, respectively, reflecting a global change of chromatin landscape between the two cell lines (Fig. 4B). Simulations again show that the dPC and FDR estimates were reasonable when $\text{SNR} > 10$ and clearly biased when $\text{SNR} < 5$ (SI Appendix, Fig. S3).

The dPC1 coefficients $\beta_{1j}$ strongly correlated with differential gene expression (DE) determined by RNA-seq (Fig. 4C; $p = 0.67$), which is an independent technology. Promoter ranking based on dPC1 predicted DE better than or as good as rankings based on each individual dataset (Fig. 4D and SI Appendix, Fig. S3D). Again, even though some datasets individually performed comparably to dPC1, in a hypothetical future application, where no prior knowledge or benchmark data are available, determining which individual dataset can provide the best ranking, and hence should be used to choose differential loci for follow-up studies, remains difficult. In that scenario, dPCA will provide a solution to integrating information automatically from multiple datasets to produce optimal or near-optimal ranking. dPC2 had a weak negative correlation with DE (SI Appendix, Fig. S3C). However, dPC1 and dPC2 jointly explained more DE than each dPC alone (SI Appendix, Fig. S3E). When promoters were grouped into nine classes based on their dPC1 and dPC2 differential states, the classes in which $\beta_{1j}$ (i.e., dPC1) and $\beta_{2j}$ (i.e., dPC2) had opposite signs had both the largest magnitude of DE (SI Appendix, Fig. S3G) and the strongest correlation between $\beta_{1j}$ and DE (Fig. 4E). This is consistent with the activation and repression nature of dPC1 and dPC2.

**Example III: Analysis of Allele-Specific Events.** ChIP-seq provides new opportunities to study allele-specific binding (ASB) and HM (22–24). ASB detection often suffers from low statistical power because only reads mapped to heterozygote SNPs contain allelic information. Also, whether or how ASB of different proteins is correlated is often unknown. One can treat the two alleles, the allele consistent with the reference genome and the non-reference allele, as paired samples from two biological conditions. dPCA can be modified to handle the paired sample data (SI Appendix, Text S1). Using the modified dPCA, we analyzed ASB in 20 ChIP-seq datasets (44 samples) from the ENCODE GM12878 cells (SI Appendix, Table S1). Genotypes for a collection of 5,504 heterozygote SNPs were obtained from a study by Rozowsky et al. (23). After removing various read mapping biases (22, 24) and applying dPCA to 2,584 bound SNPs (SI Appendix, Text S1 and Fig. S6 A–E), one dPC passed the cutoff of $\text{SNR} > 5$ (Fig. 5 A and B). This dPC is mainly driven by correlated ASB of H3K27ac, H3K4me2, H3K4me3, H3K9ac, Pol2, and c-Myc (Fig. 5A), and it positively correlates with allele-specific expression (ASE) (SI Appendix, Text S1 and Fig. S6E). At the 5% FDR level, 725 (28.1% of 2,584 and 13.2% of 5,504) SNPs were differential for dPC1. Simulations confirmed that the
dPC and FDR estimates were reasonably accurate if $SNR_j > 10$ and biased when $SNR_j < 5$ (SI Appendix, Fig. S3). In real data, $SNR_j$ was between 5 and 10. The $\nu_1$ estimate is expected to be slightly biased. This will not affect its usefulness for ranking SNPs, but the FDR estimates may be inaccurate.

We benchmarked SNP ranking in two ways. First, GM12878 is a female. Due to X-inactivation, only one allele of chromosome X (chrX) is expected to be active. Here, chrX refers to non-pseudoautosomal regions of the X chromosome. We therefore compared different ranking methods based on counting how many top-ranked SNPs were in chrX (Fig. 5C and SI Appendix, Fig. S6F). Second, using independent RNA-seq data, we obtained exonic SNPs with ASE (SI Appendix, Text S1). We compared different methods by counting how many top-ranked SNPs were in the neighborhood of exonic ASE SNPs (Fig. 5D and SI Appendix, Fig. S6F). We also did the same analysis after excluding all SNPs in chrX (SI Appendix, Fig. S6F). In all analyses, dPC1 predicted ASB better than the rankings based on individual datasets. Thus, dPCA not only allows one to explore the unknown correlation patterns of ASB across multiple proteins but improves ASB detection by using this correlation to integrate information from multiple datasets. Sometimes, the improvement can be significant.

For instance, suppose one only has the nine datasets from the Broad Institute; then, the best ranking based on individual datasets shown in Fig. 5E and SI Appendix, Fig. S5I only detected 54 chrX SNPs among the top 500 SNPs, whereas dPCA on these nine datasets detected 69 chrX SNPs (28% improvement).

Functional Interpretation and Absolute Binding. After dPCA, one may use other existing "omics" data to help with interpreting dPCs if their biological meanings are not immediately clear by looking at the $\nu_j$ patterns (SI Appendix, Text S1). For instance, in both the MYC and promoter examples, analyses of enriched gene sets were able to connect dPC1 and dPC2 to gene activation...
and repression, respectively (SI Appendix, Text S1 and Fig. S2 H and K and Fig. S5 J and M).

In all examples, we also repeated the dPCA analyses by incorporating the absolute binding information in different ways to identify differences that co-occur. This makes it very suitable for exploratory analysis of large ChIP-seq data. In principle, one may also use it to analyze other data types, such as RNA-seq. dPCA rankings of differential loci can guide design of follow-up experiments. Patterns discovered by dPCA may inform directions for improving analytical tools in various applications. For example, the correlation patterns found in the ASB analysis may provide a basis for developing new specialized tools to optimize the ASB detection power. The statistical tests in the current dPCA are based on model assumptions, such as normality and equal variances (i.e., common σ^2 for all loci and datasets), which only provide a first-order approximation to the real data. Therefore, instead of providing rigorous FDR control, the tests in dPCA often are “approximate” in nature. Empirically, this approximation worked well in our test data (SI Appendix, Text S1 and Figs. S3 and S4 E-I). In the future, the statistical tests may be improved by incorporating better data distribution assumptions tailored to specific applications.

dPCA attempts to find major patterns of differences in the data and locate the loci. Patterns with a small SNR cannot be reliably discovered, and therefore are not reported (SI Appendix, Text S1). Thus, dPCA reports main differences rather than all differences. We implicitly assume that there are some common patterns shared by many loci. If no such pattern exists, or if one wants to study loci with unique patterns, dPCA may not directly help. For detecting all loci and loci with unique patterns, a simple approach is to detect differential loci in each dataset (e.g., by t test), take their union, and find those not reported by dPCA (SI Appendix, Text S1). dPCA uses replicative variability to help with statistical inference. When there is no replicate, a variant of dPCA may be used by introducing additional assumptions (SI Appendix, Text S1 and Fig. S8). In practice, dPCs can be used or interpreted either separately or jointly depending on the available resources, and one may use other types of omics data (e.g., gene sets) to help with interpreting dPCs (SI Appendix, Text S1). Currently, absolute binding is handled by dPCA through pre-and post-processing. How to integrate the absolute binding optimally into the model to improve the analysis of differences is still an open problem worth further investigation (SI Appendix, Text S1). SI Appendix, Text S1 also includes discussions about the zero mean [i.e., E (δ_j) = 0] and equal variance (i.e., common σ^2) assumptions in dPCA. We show that these assumptions, although not perfect, are reasonable and can produce useful results.

Our data show that it is feasible to infer differential TF binding without ChIP-seq data for the TF of interest and to improve ASB analysis by exploiting correlation among multiple datasets. These examples not only demonstrate the value of dPCA but highlight the importance of developing new tools for integrative analysis of ChIP-seq data.

Methods

Data processing and analysis details for the three examples are provided in SI Appendix, Text S1. Below, we outline the dPCA algorithm and leave the mathematical details in SI Appendix, Text S1.

i) Estimate V. We first estimate σ^2 using replicate information. Then, E(Δ^2|Δ) = G = (D^2(D)^T - G) = D^2(D/J - σ^2Ω), where Diag((1/Κ1,1 + 1/Κ2,2),...,(1/Κp+p+1/umno)) is a diagonal matrix. We use the eigenvalues λ_j and eigenvectors v_j of the estimated E(Δ^2|Δ) to estimate δ_j and v_j. The proportion of variance explained by the jth dPC is computed as δ_j^2/v_j^2. Infer δ_j v_j. We have v_j^T δ_j = v_j^T δ_j + ε_j = ε_j, where ε_j ~ N(0, σ^2v_j^2). If v_j is known, one could estimate δ_j by v_j^T δ_j and test whether δ_j is zero by comparing the t-statistic T_j = v_j^T δ_j/√(v_j^T v_j). With a t-distribution, this yields a two-sided P-value P_j. In reality, v_j is unknown. Thus, we project δ_j to the estimated v_j to obtain β_j = δ_j v_j^T v_j and T_j = v_j^T δ_j/√(v_j^T v_j). We obtain the estimated P-value p_j by comparing T_j with a t-distribution. Subsequently, for each dPC p_j, s_j are converted to FDRs using the method of Storey and Tibshirani (25). Our simulations show that if the SNR for the jth dPC is big enough, the FDR computed using p_j can estimate the true FDR for testing δ_j = 0 reasonably well even if the data are projected to s_j, instead of v_j.

ii) For each pattern v_j, rank genomic loci based on |T_j|.

iv) Determine which dPCs to report. We report dPCs for which SNR = v_j^T (D^2(D/J)/G)/(σ^2v_j^2) > 5.

Acknowledgments

This research is supported by National Institutes of Health Grant R01HG006841.
Quantitative ChIP-seq data carry additional information beyond the binary binding information. Traditionally, protein-DNA binding in the ChIP-seq data is treated as a binary 0/1 event (i.e., bound vs. not bound). However, in reality ChIP-seq signals are nearly continuous rather than binary, and the continuous signals carry important quantitative information. For example, Fig. S1e shows that the H3K4me1 ChIP-seq signals obtained from the ±2kb regions surrounding the human transcription start sites (TSS) quantitatively correlate with gene expression measured by RNA-seq: the magnitude of change in H3K4me1 between two cell types, K562 and Huvec, correlates with the magnitude of change in RNA-seq (Pearson correlation = 0.44). Similarly, Fig. S1d shows that changes of H3K27ac (a histone modification) ChIP-seq signals between K562 and Huvec at MYC motif sites in the human genome quantitatively correlate with changes of MYC (a transcription factor) binding determined by MYC ChIP-seq. The quantitative nature of ChIP-seq data is reasonable since this technology is typically used to measure protein-DNA interactions (PDI) in a cell population. The varying levels of ChIP-seq signal may reflect different percentages of cells in the population that carry the PDI at different loci.

The quantitative ChIP-seq data carry important extra information beyond the binary characterization of binding. For example, a widely used approach to compare ChIP-seq data from two biological conditions is to call peaks in each condition and then draw a venn diagram using the binary peak calls. Peaks found in one condition but not in the other will be used to study differences between the two conditions. Peaks found in both conditions will be treated as “common targets”. They are usually not considered to be different between the two conditions. This binary approach has a number of limitations. First, the condition-specificity called in this way may not accurately reflect the real condition-specificity. A peak that slightly passes the peak calling cutoff in one condition but just misses the cutoff in the other condition can be called as condition-specific even if the quantitative binding levels at the peak are similar in both conditions. Second, a peak called as a common target may have dramatically different binding levels in the two conditions, and such differences cannot be detected by the binary venn diagram approach. Third, this binary approach is also sensitive to peak calling cutoffs. Changing the peak calling cutoffs can easily change the condition-specificity calls. Fourth, the binary venn diagram approach cannot be easily used to rank genomic loci within each peak category (e.g., within condition-specific peaks). Being able to rank genomic loci is important, as investigators often want to choose some promising candidates to perform follow-up experiments.

To illustrate some of the limitations of using binary peak calls to study differences between two conditions, we analyzed the ENCODE H3K4me3 ChIP-seq data from two cell types, K562 and Huvec, generated by the Broad Institute (Table S1). Using CisGenome [3], we called peaks in each cell type at the 10% false discovery rate (FDR) level. Among the peaks located within ±2kb regions of human TSSs, we identified 1447, 1848 and 10,135 K562-specific, Huvec-specific and K562-Huvec common peaks based on the venn diagram approach. For each peak, we computed the log2 fold change (log2 FC) of the ChIP-seq binding signal between the two cell types using the same data processing procedure described below for dPCA. A positive log2 FC means that K562 has stronger H3K4me3 signal compared to Huvec. We grouped peaks into six different fold change bins based on their log2 FC values: (−∞, −2), (−2, −1), (−1, 0), (0, 1), (1, 2), (2, +∞). Within each of the three peak categories (i.e., common peak, K562-specific peak, and Huvec-specific peak), we computed the frequency of peaks falling into each fold change bin (Fig. S1a). Although one would expect K562-specific peaks to have positive log2 FC, many “K562-specific” peaks called by the venn diagram approach indeed had negative log2 FC. Similarly, some “Huvec-specific” peaks called by this binary approach unexpectedly showed positive log2 FC. This clearly demonstrates a limitation of using binary peak calls to characterize differences between two conditions. Moreover, Fig. S1a also shows that “common peaks” called by the venn diagram approach can have larger |log2 FC| (i.e., absolute value of log2 FC) compared to the “K562-specific” and “Huvec-specific” peaks. For instance, some “common peaks” had |log2 FC| > 2. In contrast, a large fraction of “K562-specific” and “Huvec-specific” peaks had |log2 FC| < 2.

In Fig. S1a, the “common peak” category seemed to have a much smaller percentage of peaks in the |log2 FC| > 2 bins compared to the two condition-specific peak categories. However, recall that the total peak number in the “common peak” category was much larger than the peak numbers in the other two categories (10,135 common vs. 1447+1848 condition-specific peaks). Therefore, if one asks a different question—among all peaks with |log2 FC| > 2, what fraction are labeled as “common peaks” by the venn diagram approach—one will find that over 20% of the peaks with |log2 FC| > 2 were contributed by the “common peak” category (Fig. S1b). More over, >60% of the peaks in the 1 |log2 FC| < 2 bins were contributed by the “common peak” category (Fig. S1b). Together, these data show that using binary peak calls to characterize cell type differences will miss a significant fraction of differential sites between the two cell types.

We further examined the correlation between the differential H3K4me3 and differential gene expression determined by independent RNA-seq data from the ENCODE project (Table S1). We linked each H3K4me3 ChIP-seq peak to its closest gene using CisGenome. For each gene, we obtained the log2 FC of its gene expression levels between K562 and Huvec, using the same procedure for analyzing RNA-seq data described below for dPCA. We picked up the top 10% of the genes with the largest RNA-seq |log2 FC| values and defined them as genes with true differential expression (DE). It turned out that, among the 3295 condition-specific H3K4me3 peaks identified by the venn diagram approach, 791 (24.0%) were linked to true DE. To compare, we also used the quantitative ChIP-seq signals (i.e., |log2 FC| of the H3K4me3 binding signals) to rank all 13,430 (−1447+1848+10135) H3K4me3 peaks. Among the 3295 top peaks identified by the quantitative signals, 1179 (35.8%) were linked to DE, representing a clear improvement over the venn diagram approach (35.8% vs. 24.0%). This shows that analyzing the quantitative data allows one to better characterize between-condition differences. The binary venn diagram approach cannot be easily used to rank peaks within each peak category. If one wants to choose some differential loci for follow-up studies, one may select them randomly from the condition-specific peaks. However, the probability that a randomly chosen condition-specific peak is linked to true DE is low (24.0%). In contrast, using the quantitative data allows one to rank peaks. One can then choose the top ranked peaks for follow-up studies. When the 13,430 H3K4me3 peaks were ranked based on their ChIP-seq |log2 FC| signals, 85 (85.0%) out of the top 100 ranked peaks...
Mathematical details for dPCA

Derivation of the dPCA algorithm. The first goal of dPCA is to estimate \( V \) which is treated by us as fixed but unknown parameters. Under our model assumptions, \( \text{Var}(\delta_\beta) = E(\Delta \Sigma^2 D / \Sigma G) \) should be non-negative. Therefore if the \( m \)-th diagonal element in \( \Sigma D / \Sigma G \) is negative, we remove dataset \( m \) from the analysis by setting the corresponding row and column to zero. We then compute the eigendecomposition \( \Sigma V \Sigma^T \) for \( \Sigma D / \Sigma G \). We use \( V \) to estimate \( V \), and use \( \Lambda \) to estimate \( \Lambda \). Negative values in \( \Lambda \) are truncated to zero. The proportion of variance explained by the \( j \)-th dPC is computed as \( \lambda_j / \sum \lambda_j \). This achieves goal (1) of dPCA, except for determining which dPCs to keep.

The second goal of dPCA is to infer \( \beta_{\|g} \). We achieve this by projecting the observed data to the estimated differential patterns. In our model, we assumed that \( \beta_{\|g} \) is generated by a random mechanism through \( h_{\|g} \) and \( w_{\|g} \). In a hypothetical world, the same mechanism can be applied repeatedly to randomly generate data many times. However, in reality we only observe one realization of this process. Given the particular data we have, we treat \( \beta_{\|g} \) as fixed unknown parameters. We view \( \beta_{\|g} \) in this way in order to simplify the algorithm for implementing dPCA and to make it computationally efficient for analyzing large data. Projecting data to dPC \( \nu_j \) gives \( v_j^T d_{\|g} = v_j^T \delta_{\|g} + v_j^T \epsilon_{\|g} = \beta_{\|g} + \epsilon_{\|g} \), where \( \epsilon_{\|g} \sim N(0, \sigma^2 v_j^T \Omega v_j) \). Therefore, if \( \nu_j \) is known, one can estimate \( \beta_{\|g} \) by \( v_j^T d_{\|g} \). To test whether or not \( \beta_{\|g} \) is zero, we construct a two-statistic test \( T_{\|g} = v_j^T d_{\|g} / \sqrt{\sigma^2 v_j^T \Omega v_j} \) and compute a two-sided p-value \( p_{\|g} \) using a t-distribution with \( n \) degrees of freedom. This is based on the assumptions below:

1. \( x_{\|g} m \) is \( \mu_{\|g} + \epsilon_{\|g} m \equiv \mu_{\|g} + \lambda_{\|g} / 2 + \epsilon_{\|g} m \), where \( \lambda_{\|g} \) is a condition indicator (\( \delta_{\|g} = 1 \) if \( i = 1 \), and \( \delta_{\|g} = -1 \) if \( i = 2 \)); (2) \( \epsilon_{\|g} \)'s are independent and identically distributed as \( N(0, \sigma^2) \). We use \( M = (\mu_{\|g})_{G \times N} \) to denote the mean binding level at locus \( g \) in dataset \( m \). Recall that \( \Delta = (\delta_{\|g} \epsilon_{\|g} x_{\|g} m) \) are the true differences. Let \( \bar{x} \) be the collection of all \( x_{\|g} m \)'s, and let \( S \) be the collection of all sample variances \( s_{\|g}^2 = \sum_{m} (x_{\|g} m - \bar{x}_{\|g})^2 / (K_m - 1) \). These variances, \( s_{\|g}^2 \) are independent conditional on \( M \), \( \Delta \), and \( \sigma^2 \), and the joint probability density \( f(\bar{x}, S | M, \Delta, \sigma^2) = \prod_i \prod_m f(\bar{x}_{\|g} | \mu_{\|g}, \delta_{\|g}, \sigma^2)^2 f(s_{\|g}^2 | \sigma^2)^2 \). Here the first term \( f(\bar{x}_{\|g} | \mu_{\|g}, \delta_{\|g}, \sigma^2) \) is the density function of \( N(\mu_{\|g}, \delta_{\|g} / 2, \sigma^2 / K_m) \), and the second term \( f(s_{\|g}^2 | \sigma^2) \) is the density of \( \sigma^2 x^2 / (K_m - 1) / (K_m - 1) \). Since the observed difference \( D \) is a function of \( \bar{x} \), the independence between \( \bar{x} \) and \( S \) implies the independence between \( D \) and \( S \): \( f(D | S, M, \Delta, \sigma^2) = \prod_i \prod_m f(\mu_{\|g}, \delta_{\|g}, \sigma^2)^2 f(s_{\|g}^2 | \sigma^2)^2 \). Here \( f(\mu_{\|g}, \delta_{\|g}, \sigma^2)^2 f(s_{\|g}^2 | \sigma^2) \) is the density function of \( N(\delta_{\|g}, 1 / K_m + 1 / K_m, \sigma^2) \). Note that this function no longer depends on \( \mu_{\|g} \).

Recall that \( \sigma^2 = s^2 = \sum_{m} \lambda_{\|g}^2 / (K_m - 1) / s_{\|g}^2 / \eta \) where \( \eta = \sum_{m} \lambda_{\|g}^2 / (K_m - 1) \). Thus, \( s^2 \) is a function of \( \bar{x} \), and \( S \), and hence independent of \( D \) conditional on \( M \), \( \Delta \), and \( \sigma^2 \). Moreover, since neither \( f(\mu_{\|g}, \delta_{\|g}, \sigma^2)^2 f(s_{\|g}^2 | \sigma^2) \) contains \( \mu_{\|g} \) in the actual density function, \( D \) and \( s^2 \) are independent conditional on \( \Delta \) and \( \sigma^2 \) (i.e., the independence is not affected after integrating out all possibilities of \( \mu_{\|g} \)). As a result, if we treat \( \bar{x} \), \( \Delta \), and \( \sigma^2 \) as fixed but unknown parameters, we have (1) \( v_j^T \mu_{\|g} \sim N(v_j^T \mu_{\|g}, v_j^T \Omega v_j) \); (2) \( \sigma^2 \sim \sigma^2 x^2 / \eta \); and (3) \( v_j^T \epsilon_{\|g} \) are independent. Under the null hypothesis that \( \beta_{\|g} = 0 \), \( T_{\|g} \) follows a t-distribution with \( n \) degrees of freedom. In real applications, \( \nu_j \) is unknown. We therefore project the data to the estimated \( \nu_j \) to obtain \( \beta_{\|g} = v_j^T \nu_j^T \). We estimate the two-statistic \( T_{\|g}^2 \) by \( T_{\|g}^2 = v_j^T d_{\|g} / \sqrt{\sigma^2 v_j^T \Omega v_j} \), and then estimate the p-value \( p_{\|g} \).
by \( p_{\beta j} \) which is determined by comparing \( T_{\beta j} \) to a t-distribution with \( \eta \) degrees of freedom. Subsequently, for each dPC, we convert \( p_{\beta j} \) into FDR using the procedure described in [11]. This achieves our goal (2).

Users should be alerted that the p-values and FDRs obtained in this way are based on the model assumptions we have made. In practice, these assumptions may not fit the data perfectly. In those cases, these p-values and FDRs can only serve as an approximate rather than a rigorous measure of significance. As another note, while the errors are assumed to be normal, the effects on inference by slight deviations from the normality could more or less be alleviated when comparing sample means rather than individual observations \( (v_j^T d_p) \) is a linear combination of sample means. Later our real data analyses and simulations will show that empirically, using t-distribution as the null works well in our test data.

The third goal of dPCA is to rank differential loci to help with follow-up studies. For each estimated differential parameter \( \tilde{v}_j \), we rank genomic loci based on \( |T_{\beta j}| \). For a constant \( \delta^2 \) under the equal variance assumption, this is equivalent to ranking loci using \( |\beta_{\delta j}| \).

Finally, we determine which dPCs to report. When data are projected to the \( j \)-th dPC, one can use \( SNR_j = \text{Var}(v_j^T d_p)/\text{Var}(\hat{v}_j^T d_p) = \left(v_j^T E(D^T D/G) v_j\right)/\left(\hat{v}_j^T \hat{\Omega} d_p \right) \) to measure the signal-to-noise ratio (SNR). We estimate \( SNR_j \) using \( SNR_j = \hat{v}_j^T \hat{D}^T D/G \hat{\Sigma}_j/(\hat{v}_j^T \hat{\Omega} \hat{V}_j) \). We retain leading dPCs for which \( SNR_j \) is greater than 5. This is equal to requiring that the sampling variability contributes less than 20% of the total variation of the data when the data are projected to the \( j \)-th dPC. Our simulations show that when \( SNR_j \) is small, the estimate of \( v_j \) (i.e., the estimated direction of the \( j \)-th dPC in the \( R^M \) space) will be unreliable. Intuitively, when the noise level is high, the estimate of the eigenvector \( v_j \) can be easily rotated to point to a wrong direction. Subsequently, it is not reliable to use the estimated p-value \( p_{\beta j} \) (which is based on projecting data to \( v_j \)) to infer whether \( \beta_{\delta j} \), the projection of \( \delta^2 \) to the true \( v_j \), is different from zero or not. On the other hand, when \( SNR_j \) is big enough, FDR computed using \( p_{\beta j} \) can estimate the true FDR for testing \( \beta_{\delta j} = 0 \) reasonably well even if the data are projected to \( v_j \) instead of the true \( v_j \).

### Differences between dPCA and Factor Analysis

In dPCA, \( \text{VAV}^T \) is the eigendecomposition of \( \text{Var(}\delta^2\text{)} = E(\Delta^2 \Delta^T)/G \). If one applies PCA to the observed data \( D \), one would compute the eigendecomposition of \( \text{Var(}\delta^2\text{)} = E(D^T D/G) \). The eigenvalues from the PCA usually are bigger than the corresponding eigenvalues of dPCA (suppose all eigenvalues are arranged in descending order). To gain an intuition of why this happens, one can consider a special case where \( K_{11} = \ldots = K_{1M} = K_1 \) and \( K_{21} = \ldots = K_{2M} = K_2 \). Now all diagonal elements in \( \Omega \) are equal, i.e., \( \Omega = (1/K_1 + 1/K_2)I \) where \( I \) is an \( M \times M \) identity matrix. \( \text{Var(}\delta^2\text{)} = E(D^T D/G) = E(\Delta^2 \Delta^T/G) + \sigma^2(1/K_1 + 1/K_2)I \). Therefore the eigendecomposition of \( \text{Var(}\delta^2\text{)} \) is \( \text{V}[\text{A} + (1/K_1 + 1/K_2)\sigma^2] \text{V}^T \). This decomposition corresponds to the PCA of the data \( D \). Comparing to dPCA (applied to \( \Delta \)), the eigenvectors in PCA (based on \( D \)) remain the same, but the eigenvalues become larger because of the additional term \( (1/K_1 + 1/K_2)\sigma^2 \). This also implies that the eigenvalues from PCA will be biased estimates for the true \( \lambda_{\delta j} \). In the case above, define \( u = (1/K_1 + 1/K_2)\sigma^2 \). The percentage of variance explained by top dPCs in dPCA is \( \lambda_{\delta j}/\sum_{\delta j} \lambda_{\delta j} \). In contrast, the percentage of variance explained by the top PCs in PCA is \((\lambda_1 + u)/\sum_{\delta j} \lambda_{\delta j} + Mu)\), which is shrunk toward \( u/Mu = 1/M \). For the leading PCs, \( \lambda_j/\sum_{\delta j} \lambda_{\delta j} > 1/M \), therefore \( (\lambda_1 + u)/\sum_{\delta j} \lambda_{\delta j} + Mu < \lambda_j/\sum_{\delta j} \lambda_{\delta j} \). This explains why the leading PCs in PCA tend to explain a smaller percentage of variance in \( D \) compared to the percentage of variance in \( \Delta \) explained by the same number of dPCs in dPCA. In real applications, the diagonal elements in \( \Omega \) may not be equal. However, we still observed the same phenomenon in both real data and simulations, that is, dPCA is more efficient than PCA in terms of dimension reduction (Figs. 2a, S2g, S3d, S3(a–d–h)).

The improved dimension reduction efficiency is not the only difference between dPCA and PCA. dPCA is also different from PCA in many other important aspects. For example, dPCA integrates statistical tests with dimension reduction. It allows one to use replicate information to evaluate significance of differential loci. By separating \( \Delta \) and \( E \), dPCA also allows one to assess the potential reliability of dPC estimates based on the signal-to-noise ratio. These functions are not provided by the conventional PCA.

### Modified dPCA for analyzing paired samples

When samples from two biological conditions are paired, such as in the analysis of allele-specific events where two alleles are viewed as two conditions, we have \( K_{1m} = K_{2m} = Km \). Given \( \gamma_{g,m,k} \), the binding intensity for genomic locus \( g \), condition \( i \), dataset \( m \), and replicate \( k \), we first compute \( y_{g,m} = x_{g,m} - \bar{x}_{g,m} \). We then assume that \( y_{g,m} = \gamma_{g,m} + \epsilon_{g,m} \), and \( \epsilon_{g,m} \) are independent and follow \( N(0,\sigma^2) \). Taking average over replicates gives \( d_{gm} = \sum y_{g,m}/Km \). \( d_{gm} \) is the observed average difference between the two conditions. Organize \( d_{gm} \) into matrix \( D = (d_{gm})_{C \times M} \). We have \( D = \Delta + E \), where \( \Delta = (\delta_{g,m})_{C \times M} \) represents the unobserved true differences and \( E = (\epsilon_{g,m})_{C \times M} \) corresponds to noises. Elements in \( E \) are now independent Gaussian random variables \( \epsilon_{g,m} \sim N(0,\sigma^2/Km) \). Define \( \Omega = \text{diag}(1/K_1,1/K_1,1/K_2,1/K_2) \). After this, the dPCA algorithm introduced before can be applied to the paired sample data to estimate dPCs and perform statistical inference. Briefly, we can use \( y_{g,m} \) to estimate \( \sigma^2 \), use the eigendecomposition of \( D^T D/G - \sigma^2 \Omega \) to estimate \( \Delta \), and then use projections \( v_j^T d_p \sim N(\bar{\beta}_{\delta j}, \sigma^2 v_j^T \Omega v_j) \) to infer \( \beta_{\delta j} \).

---

**Footline Author**
Data preprocessing, normalization and transform

Basic processing. We now discuss data preprocessing for the differential TFBSs and promoter analyses. The data preprocessing for the ASB analysis will be discussed later.

Before using dPCA, reads should be mapped to a reference genome. dPCA will extend each read to 3’ end by the expected DNA fragment length in the sequencing library. This length is specified by users as a parameter for dPCA. In this article, reads uniquely mapped to human genome (hg18) were downloaded from ENCODE (http://genome.ucsc.edu/ENCODE/), and each read was extended 150 base pairs (bp).

dPCA requires users to provide the aligned reads and a list of genomic loci. Each genomic locus is a window defined by a chromosome name, a start coordinate, and an end coordinate (Fig. S1c). We use the words “locus” and “window” interchangeably in this article. For condition i, dataset m and replicate k, let N_m_k be the total number of uniquely mapped reads in the sample, and let \( q_{i,m,k} \) be the number of extended reads that overlap with genomic window \( w \). Define \( N_{\text{min}} \) to be the smallest sample read count, i.e., \( N_{\text{min}} = \min_{m,v} \{ q_{i,m,k} \} \).

dPCA normalizes and transforms the raw read count \( q_{i,m,k} \) into centered \( x_{i,m,k} \) and normalized \( x_{i,m,k} = \log_2(\frac{q_{i,m,k}}{N_{\text{min}}}) \) expression.

Next, we compute the average binding intensity for locus \( g \), condition \( i \) and dataset \( m \): \( \bar{x}_{i,m,g} = \sum x_{i,m,k}/N_{m,i} \).

The observed average difference between the two conditions is \( d_{gm} = \bar{x}_{i,m,g} - \bar{x}_{j,m,g} \). We then organize \( d_{gm} \)s into a matrix \( \mathbf{D} = (d_{gm})_{i \times m} \).

Centering column means. In dPCA, one can choose whether or not to center columns of \( \mathbf{D} \) so that each column has zero mean (centering is the default setting). Centering is done by subtracting \( d_{gm} = \bar{x}_{i,m,g} - \bar{x}_{j,m,g} \) from all \( d_{gm} \)s in each dataset \( m \). dPCA primarily considers data where raw \( d_{gm} \)s within each dataset fluctuate around zero mean (Fig. S1d,e). This case is more common than the cases where the centered \( d_{gm} \)s are very different from zero (e.g., when the binding changes in some datasets are unidirectional), applying column centering means that one needs to interpret the results in a slightly different way: the dPCs and \( \beta_{g,s} \)s characterize variations around the global mean of the differences across all loci, rather than variations around zero; the reported loci will be those whose \( \beta_{g,s} \)s are significantly different from an average locus, rather than different from zero.

Scaling column variances. Optionally, dPCA also allows one to scale columns of \( \mathbf{D} \) so that columns have equal variance. This is done by multiplying a scaling factor \( \bar{x}_{j,m,g} \) to all \( x_{i,m,g} \)s in each dataset \( m \) so that \( d_{gm} \)s in the dataset have unitary standard deviation. Users can use scaling if they believe that each dataset contributes the same amount of variance to the total data variation in \( \mathbf{D} \). By default, however, we do not scale the column variances. The data examples in the article did not use column scaling either. The reason is that column scaling may artificially increase the fraction of data variation contributed by datasets in which there is little or no difference between the two conditions. For example, Fig. S1h,i shows that the variance of \( d_{gm} \)s in the Input control dataset is smaller than the variance in the H3K27ac dataset. This is expected since the control dataset is supposed to measure background variation and is not expected to have a dynamic range as big as the H3K27ac dataset. If we scale the control dataset by a factor of 2, then a locus with \( d_{gm} = 1 \) before scaling will have \( d_{gm} = 2 \) after scaling. Since \( d_{gm} \) is in log2 scale, \( d_{gm} = 1 \) roughly corresponds to a change of two fold between the two conditions, and \( d_{gm} = 2 \) corresponds to a change of four fold. Clearly, scaling artificially amplified the differences. This is not something one wants.

Handling control datasets. In our analyses of differences, controls (e.g., Input controls) were treated as independent datasets. We did not subtract control intensities from ChIP intensities for two reasons.

First, it is quite often that multiple ChIP datasets share the same set of control samples. For example, the nine HM ChIP-seq datasets generated by the Broad Institute (BR) in the MYC analysis shared the same controls. Although one may subtract the control intensities from the ChIP intensities to obtain \( d_{gm} = (\bar{x}_{i,m,g} - \bar{x}_{j,m,g}) - (\bar{x}_{i,m,c} - \bar{x}_{j,m,c}) \) and use \( d_{gm} \) to perform analysis, subtracting the same set of controls means that \( d_{gm} \)s from different ChIP datasets (e.g., the nine HM ChIP-seq datasets from BR) will become correlated. One can no longer assume that columns of \( \mathbf{E} \) are independent. This will complicate the analysis.

Second, we observe that subtracting the control intensities from the ChIP intensities often deteriorates the quantitative nature of the signals. This is based on comparing \( d_{gm} \) (with control subtracted) and \( d_{gm} \) (without subtracting controls) in terms of their correlations with true biological signals. In the MYC analysis, for example, the Pearson correlation coefficient between H3K27ac and MYC ChIP-seq was 0.567 before subtracting the controls, and it decreased to 0.141 after subtracting the controls (Fig. S1d vs. Fig. S1f). Similarly, in the promoter analysis, the correlation between H3K4me1 and RNA-seq decreased from 0.442 to 0.316 after subtracting the controls (Fig. S1e vs. Fig. S1g). The same phenomenon was observed repeatedly in almost all datasets.

Based on these observations, we decide to treat controls as independent datasets and do not subtract them from the ChIP intensities. Although controls are viewed as independent datasets, one can still use them to assist interpretation of ChIP signals in dPCA if one wants. For instance, in the MYC analysis, the first dPC \( v_1 \) has the following \( v_{m,m} \) values for H3K27ac and control (both from BR) respectively: H3K27ac: \( v_{m,m} = 0.44 \); control: \( v_{m,m} = 0.08 \). Values in dPCA are at log2 scale. Therefore, every 10 unit increase in the coefficient \( v_{m,m} \) for dPC1 corresponds to 4.4 unit increase in H3K27ac (equivalent to a change of 21.1 folds between the two conditions) and 0.8 unit increase in the control dataset (a change of 1.7 fold), representing a net effect of 12.1 fold change if one would adjust for the control.

dPCA analysis of motif sites (Example I)

Data collection and processing. To analyze differential chromatin patterns at MYC motif sites, we downloaded 18 ENCODE ChIP-seq, DNa-seq [13, 14] and FAIRE-seq [15] datasets consisting of 70 samples from two cell types, K562 and Huvec, from http://genome.ucsc.edu/ENCODE/ (Table S1). These include 16 datasets for various chromatin marks and TFs and 2 control datasets. The data were generated by four different groups (BR: Broad Institute; UW: University of Washington; DU: Duke; NC: University of North Carolina, Chapel Hill). The uniquely aligned reads were downloaded. CisGenome [3] was first applied to call peaks at 10% FDR level for each of the 16 chromatin marks and TFs in each cell
When a protein has both IP and control samples, the two-sample peak caller in CisGenome was used. Otherwise, the one-sample peak caller was used. We did not call peaks for the two control datasets. We then mapped MYC (i.e., cMyc) consensus sequence CACCGTG to human genome (hg18) using CisGenome and obtained 138,325 motif sites by allowing no mismatch. Among them, 58,997 were covered by at least one chromatin mark or TF peak obtained above in ≥ 1 cell type. These 58,997 motif sites were analyzed by dPCA. Each motif site was extended 300 bp to both ends to create a 600 bp window. These windows were used as the input loci for dPCA.

Data preprocessing was done using the procedure described before without column centering. After the exploratory data analyses in Fig. S1, we applied dPCA to study K562-Huvec differences at these motif sites. Data for this and other examples can be downloaded from the dPCA website.

**Number of differential loci.** Fig. S2a shows the cumulative number of differential loci reported by the top $j$ dPCs. If all 18 dPCs were included, a total of 49,877 (82.9% of 58,997 and 35.3% of 138,325) differential motif sites were reported at the 5% FDR level. However, since dPC patterns with small SNR are inaccurate and have low reproducibility (see below), dPCA by default only reports the top dPCs that pass the $SNR > 5$ cutoff. In this example, only the first two dPCs passed the cutoff. Therefore, our discussions below are focused on these top two dPCs. For dPC1 and dPC2, a total of 45,507 differential motif sites were found, including 17,128 sites differential for dPC1 only, 11,473 sites differential for dPC2 only, and 16,906 sites differential for both dPC1 and dPC2 (Fig. 2c).

These 45,507 sites represented 93.1% of the 48,877 differential sites from all 18 dPCs, 77.1% of the 58,997 analyzed motif sites, and 32.9% of all 138,325 MYC motif sites.

**Evaluation of motif site ranking.** To evaluate rankings of MYC motif sites, we used independent MYC ChIP-seq data as gold standard. We downloaded aligned MYC ChIP-seq reads from the ENCODE lab at the University of Texas, Austin (UTA). These data were not included in the dPCA analysis. For each cell type, we called MYC binding peaks using CisGenome as before.

We used the dPCA data preprocessing procedure to obtain the normalized MYC binding intensities at each motif site. For each site, we computed the log2 fold change ($\log_{2} FC$) in MYC binding intensity between K562 and Huvec, using the same approach for computing $d_{gm}$. Fig. 2d in the main article shows the correlation between the MYC ChIP-seq $\log_{2} FC$ and the $\beta_{\gamma j}$ for dPC1 and dPC2 respectively. To generate Fig. 2e, we collected MYC motif sites covered by MYC binding peak in at least one cell type and with MYC ChIP-seq $|\log_{2} FC| > 1.5$ (≈ 90% of the $|\log_{2} FC|$ across all 138,325 motif sites). We obtained 64,33 gold standard differential motif sites in the genome using $d_{gm}$, which truly represents the situation where one only has information from one dataset. This method is labeled as "Single A" (i.e., rank All sites). Third, for each dataset, one can also first call peaks and identify motif sites bound by the corresponding protein in either K562 or Huvec or both, and then rank the identified sites using $d_{gm}$. We label this method as "Single P" (i.e., rank Peak sites). For clarity of display, Fig. 2e only compares dPCA with "Single PU". The comparisons with "Single A" and "Single P" are presented in Fig. S2d. In all comparisons, the ranking provided by dPC1 performed the best and outperformed rankings based on individual datasets.

Figs. 2d,e show that dPC2 only had weak negative correlation with differential MYC binding and did not predict differential binding well. This is mainly because dPC2 was driven by the repressive mark H3K27me3 which only weakly correlates with differential MYC binding (Fig. S1). The weak correlation reflects the intrinsic nature of the H3K27me3 data rather than being caused by using dPCs to combine datasets (see Text S1 “Discussion: Functional interpretation of dPCs”).

By combining dPC1 and dPC2, we were able to explain differential MYC binding slightly better. For example, among the 64,33 gold standard differential MYC binding sites, 5582 (86.8%) and 3391 (52.7%) were covered by the differential loci reported at the 5% FDR cutoff for dPC1 and dPC2 respectively (Fig. S2b).

Jointly, dPC1 and dPC2 covered 5858 (91.1%) gold standard sites, representing a larger percentage compared to each dPC alone. For each dPC, we grouped the 58,997 motif sites into three classes: (1) up in K562 (i.e., $\beta_{\gamma j} > 0$ and $FDR < 0.05$), denoted by ‘+’; (2) down in K562 (i.e., $\beta_{\gamma j} < 0$ and $FDR < 0.05$), denoted by ‘−’; (3) non-differential (i.e., $FDR > 0.05$), denoted by ‘o’. Jointly, dPC1 and dPC2 classified motif sites into nine categories. For each category, we examined the distribution of MYC ChIP-seq $\log_{2} FC$ between K562 and Huvec. It turned out that the classes in which dPC1 and dPC2 had opposite signs (i.e., the classes labeled as ‘+/−’ or ‘−/+’) on average had the largest magnitude of differential MYC ChIP-seq signal (Fig. S2e). This is consistent with the activation and repression nature of dPC1 and dPC2. When the activation module (dPC1) is high in one cell type and the repression module (dPC2) is low in the same cell type, one would expect the strongest differential binding between the two cell types.

**Evaluation of the dPC pattern and FDR estimates.** In order to evaluate whether the dPC and FDR estimates of dPCA can accurately reflect the truth, and to shed light on how many dPCs one should report, we performed simulations by keeping real data characteristics. In these “realistic” simulations, the assumptions made by dPCA may be violated in a way similar to how they are violated in real data. First, $\Delta$ was simulated using the $V$ obtained from the MYC analysis. $b_{gm}$ were simulated from Bernoulli distributions with success probability $\pi_{0} = 0.8^{+1}$ ($\tau_{0}$). $w_{\gamma j}$ were simulated as $2 \times (0.6)^{U_{gm} \times 25}$ ($U_{gm}$) were random numbers drawn from a mixture distribution 0.5&U|U&250, which mixed two t-distributions with 4 and 8 degrees of freedom with probability 0.5. One can show that for this simulation, $\lambda_{j} = (20/3) \times (0.288)^{−\tau_{0}}$. Here $\tau_{0}$ can be used to control the global signal-to-noise ratio (SNR). The $SNR_{j}$ decreases as $j$ increases. When $\tau_{0} = 0.25$, the $SNR_{j}$ in the simulation roughly matched the $SNR_{j}$ observed in the real data. After generating $\Delta$, replicate samples were created by adding noises to $\Delta$. To do so, we first computed residuals $\epsilon_{gm} = x_{gm} - \hat{x}_{gm}$ from the real data. For each sample in the real data, the residuals were organized into a vector $\epsilon_{gm} = (\epsilon_{gm}, \ldots, \epsilon_{gm})^{T}$. We created simulated samples by randomly drawing vectors $\epsilon_{gm}$ with replacement, and we used them to serve as $x_{gm}$. Next, we added the simulated $\delta_{gm}$ to $x_{gm}$ in condition 1. Simulation carried out in this way is able to retain the real data correlation structure among different genomic loci. Also, for each locus $g$, the $x_{gm}$ are...
essentially derived from its own residuals. Therefore if different loci have different variances $\sigma^2_j$, they will still have unequal variances in the simulated data (which would violate the equal variance assumption of dPCA, that is, $\sigma^2_j = \sigma^2$). Finally, since the data are simulated using real data residuals, the error distribution in real data will be retained (which may violate the normality assumption of dPCA).

We carried out the simulation using three different values of $\pi_0$ (0.1, 0.25, 0.5), corresponding to three different levels of global SNR. For each $\pi_0$, 10 independent simulations were run, and the average performance as well as the standard deviation was obtained. Figure 3 in the main article shows the simulation results for $\pi_0 = 0.25$. Results for $\pi_0 = 0.5$ and 0.1 are shown in Figure S3a,b. In all simulations, as SNR decreased, $\hat{\pi}_j$ became less and less accurate, as evidenced by the increasing distance between $\hat{\pi}_j$ and $\pi_j$ (Figures 3a,b,S3a1,a2,S3b1,b2). For dPCs with $\text{SNR}_j > 10$, the estimated dPCs generally matched the true patterns well, and the claimed FDR on average provided reasonably accurate or conservative estimates for the true FDR (Figures 3e,S3a5,a6)). The performance deteriorated as $\text{SNR}_j$ decreased (Figures 3f,S3a7,b5). For dPCs with $\text{SNR}_j < 5$, the pattern estimates became inaccurate, and the FDR estimates were off the mark (Figures 3g,S3b6,b7).

The simulation results above show that for dPCs with small SNR, the estimates $\hat{\pi}_j$ deviate from the truth (i.e., the true $\pi_j$). Moreover, Figures 3b and S3a2,b2 also show that for dPCs with small SNR, the estimates $\hat{\pi}_j$ were associated with large variability, as evidenced by the wide error bars. This implies that the estimated differential patterns for those dPCs may have low reproducibility. In other words, if two labs generate similar data independently and perform the same analysis, they may not be able to discover the same patterns for these small components. To illustrate, we randomly performed two simulations under the $\pi_0 = 0.25$ setting. The data in these simulations were generated from the same underlying true model. This mimics a situation in which two labs independently generate similar data and then apply the same algorithm to analyze the data. Figure S4a shows that the distance between the $\hat{\pi}_j$s from the two simulations increased as we moved from leading dPCs with big SNR to smaller dPCs with small SNR (see SNR in Figure 3a). For leading dPCs, the $\hat{\pi}_j$s from the two simulations were consistent. For the small dPC components, the two hypothetical labs independently generated similar data and then applied the same algorithm to analyze the data. Figure 3e shows that for dPCs with small SNRs, the $\hat{\pi}_j$s from the two simulations were consistent. For the small dPC components, the two hypothetical labs independently generated similar data and then applied the same algorithm to analyze the data. For leading dPCs, the $\hat{\pi}_j$s from the two simulations were consistent. For the small dPC components, the two hypothetical labs independently generated similar data and then applied the same algorithm to analyze the data. For leading dPCs, the $\hat{\pi}_j$s from the two simulations were consistent. For the small dPC components, the two hypothetical labs independently generated similar data and then applied the same algorithm to analyze the data. For leading dPCs, the $\hat{\pi}_j$s from the two simulations were consistent. For the small dPC components, the two hypothetical labs independently generated similar data and then applied the same algorithm to analyze the data. For leading dPCs, the $\hat{\pi}_j$s from the two simulations were consistent. For the small dPC components, the two hypothetical labs independently generated similar data and then applied the same algorithm to analyze the data.
data (see “Discussion: Functional interpretation of dPCs”). Combining dPC1 and dPC2 can explain DE better. Among the 1978 gold standard DE promoters defined by RNA-seq, 1838 (92.9%) and 1378 (69.7%) were covered by the differential promoters reported at the 5% FDR cutoff for dPC1 and dPC2 respectively. Jointly, dPC1 and dPC2 covered 1897 (95.3%) gold standard promoters, representing a larger percentage compared to each dPC alone (Fig. S5e).

We again grouped the 22368 genes into nine classes based on dPC1 and dPC2 as in the MYC example. For each class, we examined the distribution of RNA-seq log2 FC. We found that the classes in which dPC1 and dPC2 had opposite signs (i.e., the classes labeled as ‘+/−’ or ‘−/+’) on average had the largest magnitude of RNA-seq log2 FC (Fig. S5g). For each class, we also checked the Pearson correlation between ββ1 and RNA-seq log2 FC (Fig. 4e). Again, when dPC1 and dPC2 had opposite signs, the correlation between ββ1 and RNA-seq DE was the strongest. When dPC1 and dPC2 had the same sign, the correlation between ββ1 and RNA-seq DE was much weaker. This is consistent with the activation nature of dPC1 and the repression nature of dPC2. When the activation module (dPC1) is high in one cell type and the repression module (dPC2) is low in the same cell type, one can see that as the activation strength increases, there is a much weaker increase in gene expression, therefore one would expect much weaker correlation between dPC1 and DE when dPC1 and dPC2 have the same sign.

Evaluation of dPC and FDR estimates. To evaluate the accuracy of dPC and FDR estimates, we again performed realistic simulations using a procedure similar to the one described before. The simulated data retained the real data characteristics seen in the differential promoter analysis. In these simulations, ββj,s were simulated from Bernoulli distributions with success probability πj = 0.7±1−τ0. wβj,s were simulated as 2.8×(0.6)−1−Uβj where Uβj were random numbers drawn from the mixture distribution 0.5t5(u) + 0.5t5(u−1). The parameters were chosen to roughly match the SNRβ observed in the real data. The simulations were conducted using three different values of τ0 = (0.1, 0.25, 0.75). Fig. S3c–e shows the results. Consistent with the simulation results in the MYC analysis, one can see that as SNRβ decreases, the dPC estimates become less accurate, and the FDR estimates become more and more unreliable. For dPCs with SNRβ > 10, the estimated dPC matched the true pattern well, and the FDR estimates were either conservative or reasonably well (Fig. S3c(1,5,6,11,15)). For dPCs with 5 ≤ SNRβ ≤ 10, the dPC and FDR estimates were slightly less accurate (Fig. S3c(1,6,11,16,5)). For dPCs with SNRβ < 5, the dPC estimates became biased and the FDR estimates became too optimistic (Fig. S3c(1,6,11,16,5)). For dPCs with small SNRβ, the estimates ̂v were again associated with large variability (Fig. S3c(2,2,6,2)) and low reproducibility (Fig. S4b). Our simulation also confirmed that the dPC was more efficient in terms of dimension reduction compared to PCA (Fig. S3c(4,4,4,4)).

dPCA analysis of ASB (Example III)

Data collection and processing. Using data listed in Table S1 and the modified dPCA for paired samples, we analyzed allele specific histone modification and TF binding (jointly called as ASB) of multiple proteins in ENCODE Gm12878 cells. ChiP-seq reads were mapped to human genome (hg18) using MAQ [21] with default parameters. Uniquely mapped reads were extracted. For each protein, we first called binding peaks using all reads. We then obtained a collection of heterozygote SNPs in Gm12878 from [20] together with their genotypes. We treated the reference allele (i.e., the allele with the same nucleotide as the reference genome) and the non-reference allele as two conditions. More precisely, the non-reference allele was used as condition 1, and the reference allele was used as condition 2. For each SNP gj, reads mapped to each allele (i=1,2) in each dataset m and replicate k were counted. After adding a pseudo-count one to the raw counts, the counts were log2-transformed to give the xgmik.

Certain SNPs are intrinsically biased, i.e., reads from the two alleles have intrinsic differences in the ability to be mapped back to the genome. Such SNPs were removed using the procedure previously described in [22]. After applying this filter, 5504 SNPs remained. Among them, 2584 were bound by protein(s) in at least one dataset. dPCA was applied to these 2584 SNPs. Besides the intrinsic bias, there are potential reference-bias instances: given the same number of mismatches allowed in the alignments, reads from the reference allele are easier to be mapped, because reads from the non-reference allele contain one more mismatch to the reference genome (at the SNP site). To remove this reference allele bias, we first computed ygmik = xigm1k − xigm2k. For each sample (m, k), we then computed the mean ∑j ygmjk/G and subtracted this sample mean from all ygmjks in the sample. As a result, ygmbk from the same ChIP-seq sample have zero mean, making the two alleles balanced on average. We then computed ̂ymk = ∑j ygmjk/Rmk and applied dPCA to D (Figs. 5, S6).

Data exploration. dPCA reported one dPC at the SNR > 5 cutoff (Fig. 5a,b). In Fig. S6a, we also show the dPC2 pattern for readers’ reference if they want to compare it with the dPC2 in the other examples. However, dPC2 will not be reported by dPCA when used in its default mode.

After applying dPCA, we first examined the distribution of the t-statistics Tgjk for each dPC. Fig. S6b shows the histograms of Tgjk for dPC1 and dPC2 respectively. Here a negative value of Tgjk corresponds to allelic skewing to the reference allele. Therefore, if there is any reference bias, the distributions will be shifted to negative values. The plots show that for each dPC, the distribution was nearly symmetric, indicating that there is no obvious reference-allele bias. As expected, the mean of Tgjk for each dPC was zero. For dPC1, the [5, 25, 50, 75, 95]-th percentiles of Tgjk were [-5.11, -1.16, 0.19, 1.35, 4.18]. For dPC2, the [5, 25, 50, 75, 95]-th percentiles of Tgjk were [-3.49, -0.36, -0.00, 0.39, 3.53]. These data confirmed that Tgjk were not biased toward the reference allele. The box plots in Fig. S6c further show the distributions of Tgjk for all dPCs. Again, the distributions were nearly symmetric, indicating that there was no obvious reference-allele bias. We further explored the data by taking the advantage that Gm12878 is a female. Due to X inactivation, for SNPs in the non-pseudoautosomal regions in the X chromosome, only one allele is expected to be active. Below we use chrX SNPs to specifically refer to SNPs in the non-pseudoautosomal regions in chromosome X. Therefore, active chrX SNPs should be allele-specific. One may also determine which allele is active using RNA-seq data, and then use this information to study the nature of dPC1. To this end, we downloaded RNA-seq data generated by the ENCODE Yale lab for Gm12878, mapped reads and obtained allele counts at heterozygote exonic SNPs using a procedure similar to the ChiP-seq ASB analysis. For each exonic SNP, we computed the log2 fold
change (log2 FC) of RNA-seq reads between two alleles. This log2 FC can be used to measure the ASE at that SNP.

We call SNPs used in the dPCA analysis “dPCAs SNPs”. Most dPCA SNPs were not exonic SNPs. In order to link a dPCA SNP to ASE, we used CisGenome to associate the SNP with its closest gene and allowed up to 10 kb distance between the SNP and the gene’s TSS. SNPs not annotated with any gene in this way were not considered further. Not all genes have heterozygote exonic SNPs. Therefore, we also excluded SNPs from the subsequent analysis if their annotated genes did not have any heterozygote exonic SNPs. Each of the remaining SNPs was then linked to the closest heterozygote exonic SNP within its annotated gene. In this way, we were able to annotate 17 heterozygote dPCA SNPs on non-pseudoautosomal regions of chromosome X using heterozygote exonic SNPs. Subsequently, each of these 17 SNPs was linked to a RNA-seq log2 FC measuring its associated ASE. The sign of the log2 FC was adjusted based on which allele on the exonic SNP had the same parent-of-origin with the reference allele. A SNP is called “non-reference active” (NR-A) if its associated RNA-seq log2 FC is positive. Similarly, a SNP is called “reference active” (R-B) if its associated RNA-seq log2 FC was negative.

For most SNPs, the ASE log2 FC will indicate that the reference allele of the dPCA SNP is associated with active transcription. Each dPCA SNP was also associated with a βg1, the dPC1 projection value, obtained from dPCA. Again, a negative βg1 means that the reference allele has active ChIP-seq signal. Then we plotted the βg1 against the RNA-seq log2 FC (Fig. S6e). Hereinafter, we call a dPCA SNP “reference active” (R-A) if its associated RNA-seq log2 FC is negative. A SNP is called “non-reference active” (NR-A) if its associated RNA-seq log2 FC is positive.

After this adjustment, a negative value of log2 FC will indicate that the reference allele of the dPCA SNP is associated with active transcription. Each dPCA SNP was also associated with a βg1, the dPC1 projection value, obtained from dPCA. Again, a negative βg1 means that the reference allele has active ChIP-seq signal. Then we plotted the βg1 against the RNA-seq log2 FC (Fig. S6e). Hereinafter, we call a dPCA SNP “reference active” (R-A) if its associated RNA-seq log2 FC is negative. A SNP is called “non-reference active” (NR-A) if its associated RNA-seq log2 FC is positive. Similarly, a SNP is called “reference active” (R-B) if its associated RNA-seq log2 FC is positive, and a SNP is called “non-reference active” (NR-B) if its associated RNA-seq log2 FC is negative.

Fig. S6e shows that among the 17 SNPs, 8 were R-A, and 9 were NR-A. Among the 8 R-A SNPs, all 8 (100%) were R-B. Among the 9 NR-A SNPs, 88.9% were NR-B. These data show two things. First, the ASE along dPC1 and the ASE are positively correlated. For most SNPs, the ASE βg1 and the ASE log2 FC had the consistent sign. One SNP had inconsistent signs between βg1 and log2 FC. This is expected since the number of reads one can obtain for most heterozygote SNPs is small, which makes the data very noisy. Also because of the noisy nature of the data, the Pearson correlation coefficient between βg1 and log2 FC was positive but not extremely large (ρ = 0.38). However, the association between ASE and ASE was strong and statistically significant. A Fisher’s exact test of association using the 2 x 2 contingency table [0.8:8.1] gave a two-sided p-value = 0.0004.

Second, there was no obvious reference bias, since the four counts [0.8:8.1] were almost symmetric. Let r1 be the percentage of R-A SNPs that were R-B, and r2 be the percentage of NR-A SNPs that were NR-B. If there is reference bias, one would expect that r1 and r2 should be different, and r1 > r2. However, a Fisher’s exact test using the 2 x 2 contingency table [8.1:0.8] gave a two-sided p-value = 1, suggesting that the two percentages were not significantly different.

Evaluation of SNP ranking. We used two methods to evaluate SNP rankings. First, taking the advantage that only one allele of the chrX SNPs (defined as SNPs from the non-pseudoautosomal regions of chromosome X) is expected to be active in Gm12878, we identified chrX SNPs among the bound and analyzed SNPs and treated them as true positives. Different methods were compared by counting the number of true positives among the top ranked SNPs (Figs. 5c, S6f). In principle, one may further annotate chrX SNPs using ASE to get a more stringent gold standard set. However, this would result in very small number of gold standard SNPs which is not enough for robustly comparing different methods. Therefore, we did not further pursue this approach.

Second, we counted the number of SNPs associated with allele-specific expression (ASE) among the top ranked SNPs. Using the RNA-seq data described above, we computed the log2 FC of RNA-seq reads between the two alleles for all heterozygote exonic SNPs in the genome. The heterozygote exonic SNPs with at least 6 aligned reads and |log2 FC| > 1 were defined as ASE SNPs. We annotated SNPs in the dPCA analysis with their closest genes using the procedure described above, and defined a SNP to be associated with a ASE gene if the SNP’s annotated gene has an exonic ASE SNP based on RNA-seq. Using this criterion, we evaluated how many top ranked SNPs in the dPCA analysis are associated with ASE, and compared dPCA rankings with the rankings based on individual datasets (Figs. 5d, S6g).

Results from both evaluation methods indicate that dPC1 performed the best in terms of predicting allele-specific events. For the analysis using RNA-seq as benchmark, we also repeated the analysis using autosomal SNPs only (Fig. S6h), and the result holds the same.

Finally, by applying similar procedures, we also performed dPCA ranking analysis using only the nine datasets from the Broad Institute (Figs. 5e, S6i-k). Now the improvement of dPCA over individual dataset analyses in terms of SNP ranking became even stronger (e.g., Figs. 5c vs. 5e). This illustrates that in some applications involving ASB inference, dPCA may result in significant improvement over the simple approach that analyzes datasets separately.

Evaluation of dPC and FDR estimates. To evaluate FDR estimates, we performed realistic simulations. Here we simulated 8y = xgmk − xgmk rather than xgmk − βyg1 was simulated as 1.6 * (0.5)^0.5 * (0.1), where Uyg1 was random numbers drawn from the mixture distribution 0.5*Uu + 0.5*Us (0.5, 0.25, 0.1). Fig. S3f-h shows the results. Consistent with previous simulation results, the FDR estimates were reasonable when SNR1 = big, and became biased when the SNR2 decreased. In the real data, the SNR2 for dPC1 was between 5 and 10. In that range, the estimate of τ1 = slightly biased in the simulation, but the FDR estimates were not accurate.

Assessment of t-distribution as a null. Our current model uses t-distribution as the null distribution. There are multiple ways to check whether it is reasonable to use this null in real data which will be illustrated below.

First, one may check the normality of residuals εgmk = xgmk − 8y, using QQ-plot. Fig. S4e shows a few example QQ-plots from our analyzed data. Here despite some deviations, the normality assumption is not a bad first order approximation to use.
Second, as a way to check the independence between the numerator and denominator in the t-statistic, one may plot residuals $g_{m,k}$ versus $d_{m}$ in each dataset $m$. See Fig. S4f for a few examples in which there was no significant correlation between $\epsilon$ and $d$. Note that many plots in Fig. S4f have symmetric residuals because their corresponding datasets have only two replicates, and hence the residuals occur in pairs with opposite values $\epsilon$ and $-\epsilon$.

Third, one may also want to further check the independence between $v_j^2d_{j}^2$ and $s^2 = s^2$ directly. Since $s^2$ is a single number, one cannot directly use $s^2$ to check the independence. However, one can plot the $s_g = \sqrt{\sum_m \sum_k (d_{g,m,k} - \bar{d}_g)^2 / \sum_m \sum_k (R_{m,k} - 1)}$ or $s_g = \sqrt{\sum_m \sum_k (y_{g,m,k} - \bar{y}_g)^2 / \sum_m (K_{m} - 1)}$ in the paired sample case versus $\beta_{g,j} = v_j^2d_{j}^2$ for each reported dPC (Fig. S4g). Fig. S4g shows that in the data we analyzed, there was no strong dependence between $s_g$ and $\beta_{g,j}$. For each plot in Fig. S4g, we divided genomic loci into $L = 100$ for MYC and Promoter, and 10 for ASD strata based on $\beta_{g,j}$ values. Each stratum contained the same number of loci. Loci in the same stratum had similar $\beta_{g,j}$ values. For each stratum $l(= 1, \ldots , L)$, we computed the sample mean, standard deviation, and the $5^{th}$, $25^{th}$, $50^{th}$, $75^{th}$ and $95^{th}$ percentiles of $s_g$. For each type of statistic (e.g., sample mean), this yields $L$ different values. We computed the coefficient of variation (CV) of these $L$ values. Fig. S4h shows the CVs for different statistics. All CVs are very small compared to one, indicating that these summary statistics do not vary much compared to their mean value when $\beta_{g,j}$ changes. Thus there is no clear dependence between $s_g$ and $\beta_{g,j}$. Since $s^2$ is a linear combination of $s^2$, the independence between $\beta_{g,j}$ and $s_g$ would imply the independence between $\beta_{g,j}$ and $s^2$. Note that checking the normality using QQ-plots in Fig. S4e may also help here since the normality and independence of errors imply the independence between $D$ and $s^2$.

Fourth, if the model assumptions are correct, p-values for each dPC should follow a mixture of uniform distribution (for true null hypotheses) and a distribution skewed toward 0. Fig. S4i shows that in all our real data examples, the right tail of the p-value distribution is nearly uniform, showing no alerting sign for serious problems in model assumptions.

Finally, one may use simulations that keep real data residuals to assess the FDR estimates based on the t-distribution (e.g., Fig. S3). Even though the real data residuals may not perfectly satisfy our model assumptions such as normality and equal variance, Fig. S3 shows that using t-distribution was still able to provide reasonable FDR estimates in our test data. Besides the simulations in Fig. S3, we also performed simulations in other ways and obtained very similar results. In the interest of space, those additional simulations are posted on the dPCA website.

**Applying dPCA to binding peaks only (dPCA-P).** dPCA takes a list of genomic loci as input. This design gives users the flexibility to specify which genomic loci to analyze. Therefore, users may use any existing ChIP-seq peak caller to identify binding locations in each dataset and then take the union of the binding locations from all datasets as the input for dPCA. This approach can exclude genomic loci without significant binding from the dPCA analysis.

For user’s convenience, the dPCA package contains a function to support this preprocessing step. To use this function, users first provide a list of genomic loci and aligned sequence reads for all samples, and specify the experimental design (i.e., which samples are the IP and control samples in each ChIP-seq experiment). The function will then use the peak caller in CisGenome to detect binding locations for each protein in each condition using 10% FDR as the default cutoff. If an experiment has both IP and control samples, the two-sample peak caller in CisGenome will be used to find peaks. This allows users to use the control samples to remove non-specific binding. If an experiment only has IP samples without any accompanying control sample, the one-sample peak caller in CisGenome will be used. Using the called peaks, each genomic locus $g$ provided by the users will be labeled as bound ($c_{g,m} = 1$) or not bound ($c_{g,m} = 0$) for condition $i$ and dataset $m$ based on whether the locus overlaps with any peak in the corresponding dataset. Users can then apply dPCA to loci that are bound by at least one protein in at least one biological condition. In other words, let $c_{g,m} = \max(c_{g,m,1}, c_{g,m,2})$, and set $c_{g,m} = 0$ if $m$ is a control dataset. dPCA can be applied to loci for which $\sum_m c_{g,m} > 0$. This approach was used to perform the analyses in the main article.

**Filtering differential loci using $R^{BD}$.** Recall that $c_{g,m} = \max(c_{g,m,1}, c_{g,m,2})$ indicates whether locus $g$ is bound in dataset $m$ in at least one condition. Also, recall $v_j^2d_{j}^2 = \sum_m v_j^2d_{j,m}$. The $R^{BD}$ statistic for genomic locus $g$ and dPC $j$ is defined as

$$R^{BD}_g = \frac{\sum_m c_{g,m}(v_jd_{j,m})^2}{\sum_m (v_jd_{j,m})^2}$$

1

Here $B$ stands for Binding. $(v_jd_{j,m})^2$ is the squared difference along dPC $j$ contributed by dataset $m$. The denominator $\sum_m (v_jd_{j,m})^2$ represents the total between-condition variation along dPC $j$. The numerator $\sum_m c_{g,m}(v_jd_{j,m})^2$ is the amount of variation covered by binding peaks. Thus for a locus $g$ and dPC $j$, $R^{BD}_g$ measures the fraction of between-condition variation that co-occurs with significant absolute binding. $R^{BD}_g = 0.9$ means 90% of the between-condition variation observed at locus $g$ along dPC $j$ are associated with significant binding peaks. In $R^{BD}_g$, $d_{j,m}$ is weighted by the dPCA coefficient $v_j$. In order to have a large $R^{BD}$ for a particular locus $g$ and dPC $j$, it is therefore important to have peak calls at the key datasets that drive the differences (i.e., datasets with large $v_j$ and $d_{j,m}$ in terms of magnitude).

Users can apply the $R^{BD}$ filter after dPCA-P and select differential loci with large $R^{BD}$ values (e.g., $R^{BD} > 0.5$) for further study. As a post-processing technique, $R^{BD}$ is different from the filter used in dPCA-P. dPCA-P retains loci bound in at least one dataset, but the binding peak(s) may not occur in the same datasets in which the main between-condition differences occur. $R^{BD}$ can more specifically identify loci where the differences and binding peaks co-occur in the same datasets.

When we implement $R^{BD}$, $c_{g,m}$ is actually computed as $c_{g,m} = 1 - \text{FDR}_{g,m}$. Here $\text{FDR}_{g,m}$ is the FDR of the binding peak associated with locus $g$ in condition $i$ and dataset $m$. It is produced by the peak calling function in dPCA. $\text{FDR}_{g,m}$ is set to 1 if no peak is reported by the peak caller for locus $g$. 

---

**Incorporating absolute binding information**

The basic dPCA model analyzes differences at any loci that cannot be explained by cross-sample variation. Such differences may occur at locations where the binding (also called “absolute binding”) level is low. Differences not associated with significant binding may not be easy to study experimentally, and interpreting them could also be challenging since it is not easy to tell whether they are real biological differences or differences in non-specific binding or backgrounds not captured by the cross-repeatation variation. For these reasons, we provide multiple options for users to focus on analyzing differences at locations where there are significant binding.

Footline Author

PNAS | Issue Date | Volume | Issue Number | 9
condition $i$ and dataset $m$ at the 50% FDR cutoff. Therefore, strictly speaking, $c_{gm}$ is a number between 0 and 1 rather than a binary indicator in our implementation. In this way, peaks with smaller FDRs will bring relatively higher weight $c_{gm}$ to $w_{gm} = (c_{gm}w_{gm})^{2}$ when we compute $R_{gm}$.

In the ASB analysis, the computation of $R_{gm}$ is slightly different from the MYC and promoter analyses. For inferring allele-specificity, only a small fraction of reads that are aligned to heterozygote SNPs are informative. Therefore, $d_{gm}$ is based on reads mapped to heterozygote SNPs. For identifying binding locations, however, there is no need to limit oneself to the small number of reads mapped to the heterozygote SNPs. We therefore use all reads to find peaks. Instead of having two allelic peak calls $c_{g1m}$ and $c_{g2m}$, we only have one pooled peak call $c_{gm}$ at each SNP which will be used along with $d_{gm}$ to compute the $R_{gm}$ statistic as before.

Below we illustrate $R_{gm}$ using the MYC analysis as an example. We used the dPCA peak calling function described before to compute $c_{gm}$ and $R_{mg}$. Figure S7a(1-5) shows a few loci differential along dPC1 with different $R_{gm}$ values. At these loci, dPC1 activity is higher in Huvec compared to K562. For each locus, the left panel shows the normalized binding intensities $x_{g1m}$ and $x_{g2m}$, in K562 and Huvec in each dataset $m$. If a dataset has a binding peak call, a star is shown on top of that dataset. In the right panel, we show the MYC binding intensities in the benchmark MYC ChIP-seq data. A gold star is shown if the locus has differential MYC ChIP-seq $|\log_{2}(FC)| > 1.5$. The locus in Figure S7a(1) has $R_{g1m} = 0.99$. It has high binding level in all datasets that drive dPC1 (i.e., H3K4me1, H3K4me2, H3K4me3, H3K9ac, H3K27ac). Differences at this locus are associated with differential MYC binding (i.e., gold star). The locus in Figure S7a(2) has $R_{g2m} = 0.4$. The binding level is lower. However, the locus still has binding peaks in the H3K4me1 and DNase-seq data. Its binding intensities in several other datasets that drive dPC1, including H3K4me2, H3K9ac and H3K27ac, did not pass the peak calling cutoff, but were still higher than the background represented by the Input control dataset. This locus is also a differential MYC binding site. The loci in Figure S7a(3,4) have small $R_{gm}$ ($R_{g2m}$ = 0.01 and 0.03 respectively). Despite the observed K562-Huvec differences, these loci have low absolute binding level in most datasets that drive dPC1. The locus in Figure S7a(4) is associated with differential MYC binding, but the locus in Figure S7a(3) is not. The locus in Figure S7a(5) has lower rank compared to the loci in Figure S7a(2-4). It has smaller K562-Huvec difference but strong absolute binding and therefore a large $R_{gm}$ ($R_{g1m} = 0.77$). It is not differentially bound by MYC. Figure S7a(6-8) shows a few differential loci along dPC2. Again, large $R_{gm}$ is associated with high binding level in the datasets that drive dPC2.

Next, we collected all differential loci reported by dPCA at the 5% FDR level. For each dPC, the differential loci were grouped into four bins based on their $R_{gm}$ values: $R_{gm} \in [0, 0.25]$, $(0.25, 0.5]$, $(0.5, 0.75]$ and $(0.75, 1]$. For each bin and each dataset, we asked what percent of the differential loci was associated with binding peak calls in that dataset. Figure S7b shows that in the datasets that mainly drive each dPC, increasing $R_{gm}$ usually increases the peak call percentages. For instance, H3K4me1, H3K4me2, H3K4me3, H3K9ac and H3K27ac are the drivers for dPC1. In these datasets, differential loci with $R_{g1m} > 0.75$ had much higher peak call percentages compared to loci with $R_{gm} \leq 0.25$. Note that the peak call percentages in this figure were all below 100%, even for the most stringent bin with $R_{gm} > 0.75$. This is reasonable since different datasets may have different quality and signal-to-noise ratio, and as a result, some true binding events may not pass the peak calling cutoff in some datasets. In addition, biologically meaningful differences may occur along a dPC even if only a subset of the datasets has significant binding peak calls (e.g., see Figure S7a(2)). For these reasons, we usually do not require a differential locus to have binding peak calls in all driving datasets in order to be considered to have significant absolute binding along a dPC.

We also examined the binding quantitatively for loci in different $R_{gm}$ bins. To this end, we used $a_{gm} = x_{g1m} + x_{g2m}$ to measure the amount of binding at locus $g$ in dataset $m$. For each dPC, the differential loci were stratified based on their dPCA ranks. Each stratum contained 1000 loci with similar ranks. Loci within each stratum were then grouped into four $R_{gm}$ bins as before. Within each bin, the mean of $a_{gm}$s across all loci was computed for each dataset $m$. Figure S7c,d shows the results from two representative strata: rank $\in (9000,10000]$ (Figure S7c) and $(25000, 26000]$ (Figure S7d). These figures show that for peaks with similar ranks (and therefore similar amount of differences), large $R_{gm}$ usually tends to have higher absolute binding level across all datasets in the driving datasets for each dPC. We note that loci in Figure S7d were ranked lower than loci in Figure S7c. However, comparing loci with $R_{gm} > 0.75$ from dPC1 in these two figures, loci in Figure S7d tend to have larger $a_{gm}$. This is because many higher ranked differential loci tend to have strong binding in one condition but weak binding in the other (this configuration often yields strong differences), whereas many lower ranked loci had high binding level in both conditions. As a result, $a_{gm} = x_{g1m} + x_{g2m}$ can be higher for the lower ranked differential loci. Instead of grouping loci into strata, for each dPC we also pooled all differential loci together and examined the average $a_{gm}$ in the four $R_{gm}$ bins. Large $R_{gm}$ again correlates with high binding level. Due to the space limit, this result is posted on the dPCA website (http://www.biostat.jhsph.edu/~hji/dpc/anals/FigureWeb1.pdf).

Analyses of the Promoter and ASB data produced similar results, which are also posted on the dPCA website (http://www.biostat.jhsph.edu/~hji/dpc/anals/FigureWeb1.pdf). Together, our analyses show that $R_{gm}$ offers a solution to selecting differential loci whose differences co-occur with significant absolute binding.

We explored distributions of $R_{gm}$ among differential loci in the three real data examples. For each dPC, differential loci were grouped into four $R_{gm}$ bins as before. Figure S7e shows the percentage of differential loci in each bin among the top $N$ ranked loci. The percentages are shown as a function of $N$. For instance, in the MYC analysis, 96.0% (9604) of the top 10000, and 89.4% (17871) of the top 20000 differential loci along dPC1 had $R_{gm} > 0.5$. In all three examples, top ranked loci were more likely to have large $R_{gm}$. This is reasonable since top differential loci are often associated with big differences. Intuitively, a strong difference $d_{gm} = x_{g1m} - x_{g2m}$ means one of $x_{g1m}$ and $x_{g2m}$ is big and therefore has high probability to correspond to a binding event. Thus the top ranked loci tend to have large $d_{gm}$ and binding peak calls at the same time, and subsequently large $R_{gm}$. We wish to point out, however, that this trend is not guaranteed to hold true in all future data. For example, if all genomic loci analyzed by dPCA have binding peaks in all datasets, then all loci will have the same $R_{gm} = 1$. We also note that despite the observed correlation between $R_{gm}$ and the locus rank produced by dPCA, they contain different information and are not redundant. For instance, the locus in Figure S7a(4) was ranked higher than the locus in Figure S7a(5), but the higher ranked
locus had smaller \( R^2 \). From Fig. S7a(1-3), one can also see that loci with similar ranks can have very different \( R^2 \) values.

In the MYC analysis, among the differential loci reported at the 5% FDR level, 73.6% (25044/34034) and 64.2% (18205/28379) had \( R^2 > 0.5 \) for dPC1 and dPC2 respectively. In the promoter analysis, these percentages were 91.3% (15519/16990) and 51.0% (6998/13735). In the ASB analysis, 92.0% (667/725) of the differential loci reported for dPC1 had \( R^2 > 0.5 \). In both the MYC and promoter data, the percentage of loci with large \( R^2 \) values was lower for dPC2 compared to dPC1. This is mainly because H3K27me3 in these data had weaker signals and fewer binding peak calls overall compared to the gene activation marks that drive dPC1. For instance, among all the 58997 analyzed motif sites in the MYC example, 61.1% were associated with H3K4me1 binding, whereas only 32% and 41.5% were associated with H3K27me3 in the BR and UW datasets respectively.

Resetting differences at non-binding loci to zero (dPCA-Z).

This is an ad hoc approach which sets \( d_{\text{gm}} \) to zero before dPCA if locus \( q \) is not covered by any peak in dataset \( m \) (i.e., if \( c_{\text{gm}} = 0 \)). All the other \( d_{\text{gm}} \)s will remain unchanged. This will create a modified matrix \( \mathbf{D} \) in which differences not associated with binding peaks all become zero. Similar to dPCA-P, loci not bound in any dataset (i.e., \( \sum m c_{\text{gm}} = 0 \)) are excluded from the dPCA-Z analysis. In dPCA-Z, we only change \( d_{\text{gm}} \) and do not change the residuals (i.e., \( x_{\text{gm}} - \hat{x}_{\text{gm}} \), at each locus will remain unchanged). In other words, the sample variance \( s^2 \) and the \( \sigma^2 \) estimate will remain the same compared to dPCA-P. dPCA-Z runs the dPCA algorithm using \( \mathbf{D} \) and \( \sigma^2 \). After finding \( \hat{\mathbf{v}}_j, \hat{\beta}_j \) is computed as \( \mathbf{v}_j^T \mathbf{d}_j \). T-statistics are constructed according as \( T_{\text{gjm}} = \mathbf{v}_j^T d_j / \sqrt{s^2 \mathbf{v}_j^T \Omega \mathbf{v}_j} \). dPCA-Z allows one to more specifically analyze patterns of differences using data only from binding peaks. A potential drawback of this approach is that it can artificially change the underlying covariance structure of the data as it forces some \( d_{\text{gm}} \)s to become zero. We note that the binary peak calls do not represent a perfect description of the continuous binding signals. Therefore, it is possible that \( d_{\text{gm}} \)s not associated with binding peaks can represent weak binding but true differences (e.g., see H3K4me2 in Fig. S7a(2)). These differences will be removed after being set to zero. In dPCA-Z, differences not covered by any peak have zero contribution to \( \mathbf{v}_j^T \mathbf{d}_j \), therefore all reported differences are guaranteed to come from places where there are binding peaks. For this reason, if dPCA-Z is used, we will not use \( R^2 \) to further filter differential loci.

Comparison of different methods.

We tested the methods below for incorporating the absolute binding information.

- **P**: This is the dPCA-P approach described above. One example is the analysis of the 58997 MYC motif sites associated with at least one chromatin mark.
- **PR**: After applying dPCA-P, differential loci are further filtered using \( R^2 \). Loci with \( R^2 \leq c \) are eliminated. Three different cutoffs, \( c = 0.25, 0.5 \) and 0.75 were tested. Results for these three settings are labeled using PR.0.25, PR.0.5 and PR.0.75 respectively.
- **A**: dPCA is applied to all genomic loci regardless of their binding status. One example is to apply dPCA to all 138,325 MYC motif sites in the genome.
- **AP**: After applying dPCA to all loci (A), differential loci not covered by any binding peak (i.e., loci with \( \sum_m c_{\text{gm}} = 0 \)) are filtered out. The filter here is similar to P, but \( \hat{\beta} \) filters out loci before dPCA, whereas AP filters out the same loci after dPCA.
- **APR**: After AP, differential loci are further filtered using \( R^2 \). Again, three cutoffs were tested. They are labeled as APR.0.25, APR.0.5 and APR.0.75 respectively.
- **Z**: This is the dPCA-Z approach.

First, we compared dPC patterns reported by different methods. dPCs discovered by A (apply dPCA to all loci) and Z (dPCA-Z) are shown in Fig. S4j-l. They are largely similar to patterns discovered by P (dPCA-P) (Figs. 2a,4a,5a). This shows that the patterns discovered by dPCA from these data are very stable: using the most relaxed loci list in A or the highly stringent pre-processing in Z produced dPCs similar to those in dPCA-P, suggesting that the reported dPCs represent intrinsic differential patterns in the data. Filtering loci using \( R^2 \) is a post-processing step, therefore P and PR will always report the same patterns. Similarly, A, AP and APR will always report the same patterns.

Second, we compared the loci rankings of different methods. Fig. S7f shows the number of gold standard, \( X_N \), found in the top \( N \) ranked loci in each method. The figure also contains the results from the single dataset ranking including single A (i.e., rank all genomic loci, e.g., all 138325 MYC motif sites, in each dataset using \( d_{\text{gm}} \)). Single P (i.e., rank all loci bound in each dataset by \( d_{\text{gm}} \)) and “Single PU” (i.e., rank the union of loci bound in \( \geq 1 \) dataset, e.g., the 58997 bound MYC motif sites, using \( d_{\text{gm}} \)). Fig. S7g plots the number of gold standard \( X_N \) versus \( 1 - X_N / N \). Here \( X_N / N \) is the Positive Predictive Value (PPV) and \( 1 - X_N / N \) is the percentage of false positives if the gold standard is viewed as true positives. The curves in Figs. S7f and S7g can be converted into each other, and therefore carry the same information except that we did not show the single dataset results in Fig. S7g to allow a clear visual comparison among different dPCA variants. In all plots, different methods are represented by curves with different line styles and colors. In the MYC and promoter plots, there are two sets of colored curves for dPCA. The one on the upper part of each plot corresponds to dPC1. The one on the lower part of each plot corresponds to dPC2. Figs. S7f-g show that applying dPCA to all loci (A) and to the union of peak loci (P) performed similarly in the promoter data, but A did not perform as well in the MYC and ASB data. Comparing P, PR.0.25, PR.0.5 and PR.0.75, we observe that using the \( R^2 \) filter did not clearly improve the ranking compared to P. In fact, PR.0.75 performed even worse in the MYC data. Therefore, although applying the \( R^2 \) filter after dPCA-P allows one to more stringently define differential loci by requiring a higher level of absolute binding, it does not necessarily result in better ranking performance in terms of predicting the gold standard.

In reality, it can happen that a more stringent method has lower prediction performance. For instance, increasing stringency could result in decreased sensitivity which could then affect the overall receiver operating characteristics. To demonstrate, consider dPC1 in the MYC analysis. Fig. S7h shows the PPVs of P, PR.0.25, PR.0.5 and PR.0.75 as a function of the t-statistic’s \( -|T_{\text{gjm}}| \). In other words, for each t-statistic cutoff \( t \), we collected top loci with \( |T_{\text{gjm}}| > t \), or equivalently \( -|T_{\text{gjm}}| < -t \), from each method. We used \( -|T_{\text{gjm}}| \) in the plot so that the top ranked loci will be on the left, consistent with Fig. S7f. Fig. S7h then shows the PPV of the top ranked loci with \( -|T_{\text{gjm}}| < -t \) for each method. Recall that \( T_{\text{gjm}} \) is the absolute value of \( T_{\text{gjm}} \) from the four methods were obtained by projecting data to the same vector \( \mathbf{v}_j \), therefore they can be directly compared: loci with similar \( |T_{\text{gjm}}| \) have similar amount of differences. The figure shows that for a fixed \( t \), loci with larger \( R^2 \) were associated with larger PPV. This suggests that for loci with similar amount of differences, those with higher absolute binding de-
fined by $R^2$ are more likely to be true positives. However, when one looks at Fig. S7f which shows the number of gold standard sites detected by each method among the top ranked loci, one can see that increasing $R^2$ also reduced the sensitivity: fewer gold standard sites were detected.

Figs. S7a(2,4) show two example loci which were true MYC differential binding sites but with $R^2 < 0.5$. These loci were filtered out by applying a stringent $R^2$ cutoff. The ranking results in Figs. S7f,g represent the net effect of applying $R^2$ after considering both the sensitivity and specificity. Instead of fixing the t-statistic cutoff $t$, Fig. S7f compares different methods by fixing the number of reported loci. Intuitively, in order to report the same number of loci using P and PR, one needs to have lower t-statistic cutoff $t$ for PR to include additional loci to compensate for those filtered out by $R^2$. These newly added loci may not necessarily have a lower false positive rate compared to the loci filtered out by $R^2$. Therefore, the net effect of the sensitivity-specificity tradeoff is that applying the $R^2$ filter did not improve the overall ranking performance. As shown by the two loci in Fig. S7a(2,5), a locus ranked higher in dPCA-P with a smaller $R^2$ could be a true differential MYC binding site, whereas a locus ranked lower but with larger $R^2$ may not necessarily correspond to a true differential MYC binding site.

Even though applying the $R^2$ filter did not improve the ranking performance, Figs. S7a-d show that it does allow one to identify differential loci that also have significant absolute binding which many investigators may want to collect for various reasons. For instance, it may be easier to experimentally validate the binding at these loci, and owing to limitations of the technology and knowledge, it may not be easy to experimentally verify and interpret the differences at locations with weak binding even if many of them could be biologically meaningful differences. We therefore provide the option to compute $R^2$ values for each differential locus. Users can choose to use the $R^2$ filter or not depending on their needs.

Figs S7f,g also show that AP and APR performed largely similarly to P and PR. Thus filtering out the non-peak loci before dPCA or after dPCA did not result in dramatically different ranking performance. Finally, Z (dPCA-Z) did not bring clear improvement over P either. In fact, Z even performed slightly worse in the MYC data.

Third, we compared the number of differential loci reported by different methods in Fig. S7j. For each data example, the figure shows two bar plots. The left plot shows the number of loci reported by each method. The right plot shows the number of gold standard loci detected by each method. Within each plot, each dPC has three horizontal bars. The bar on the top shows how the total number of differential loci increases as one moves from PR0.75 to PR0.5, PR0.25 and P, respectively. The bar in the middle shows how the number of loci increases as one moves from APR0.75 to APR0.5, APR0.25, AP and A, respectively. The bar on the bottom shows the number of loci reported by Z. For the ASB data, the gold standard loci here were based on RNA-seq ASE. The results based on chrX gold standard were similar and therefore not shown here due to the space limit. These plots show that decreasing $R^2$ in the PR and APR often resulted in inclusion of more differential loci and more gold standard sites. Therefore, if the goal is to comprehensively discover more gold standard sites (i.e., to achieve high sensitivity), one may want to use P or AP without applying the $R^2$ filter. Compared to AP and P, A often reported more differential loci without significantly increasing the number of detected gold standard sites. The number of loci obtained from APR0.75, APR0.5, APR0.25 and AP were similar to those obtained from PR0.75, PR0.5, PR0.25 and P, respectively. The number of loci reported by Z was usually between the numbers from PR0.75 and PR0.5.

Fourth, we further compared the consistency of the loci reported by different pairs of methods in Fig. S7k. For each data example, dPC, and pair of methods X and Y (written as X-Y in the figure), the figure shows two bars. The top bar is the percentage of loci reported by X that are also discovered by Y, both at the 5% FDR level. The bottom bar is the percentage of loci reported by Y that are also discovered by X. The figure shows that loci reported by APR0.75, APR0.5, APR0.25, AP were highly consistent to those reported by PR0.75, PR0.5, PR0.25 and P, respectively. Most loci reported by Z were contained in those reported by PR0.5 and APR0.5, whereas most loci reported by PR0.75 and APR0.75 were contained in those reported by Z. This indicates that the filtering effect of Z was between [PR/AP]0.5 and [PR/AP]0.75.

Finally, we checked how filtering differential loci using the absolute binding may influence the results in various downstream analyses. We repeated various analyses using the differential loci defined by five representative methods: A, P, PR0.5, PR0.75 and Z. These methods covered different stringency levels of the absolute binding filter. Regardless of which method was used, all the major observations reported in the main article which were based on P (dPCA-P) remained qualitatively the same. For instance, Figs. S2b,c,S5e,f show the numbers of differential loci reported by dPC1 and dPC2 in the MYC and promoter examples respectively. For all methods, dPC1 and dPC2 jointly covered more differential loci and more gold standard sites than each dPC alone. Figs. S2f,S5h show that regardless of the methods, the differential gold standard signal was the strongest when dPC1 and dPC2 had opposite signs. Here in PR0.5 and PR0.75, we used $R^2$ to more stringently classify loci analyzed by dPCA-P into three classes along each dPC. For instance, in PR0.5, we defined dPC1+ using $FDR < 0.05$, $g_3 > 0$ and $R_{31}^g > 0.5$. dPC1- was defined using $FDR \leq 0.05$, $g_3 < 0$ and $R_{31}^g > 0.5$. dPC10 was defined using $FDR > 0.05$. Loci with $FDR \leq 0.05$ but $R_{31}^g \leq 0.5$ were ambiguous and did not belong to any class. Similarly, we more stringently defined dPC1(+/-)-dPC2(+/-) by requiring $FDR \leq 0.05$ and $R_{31}^g > 0.5$ for both dPCs. dPC1(+/-)-dPC20 was defined by requiring $FDR < 0.05$ and $R_{31}^g > 0.5$ for dPC1, and $FDR > 0.05$ for dPC2. Other patterns were defined in a similar way. In A and Z (dPCA-Z), we grouped the loci analyzed by each method into different dPC classes based on FDR and the sign of $g_3$, similar to how it was done in P (dPCA-P). Also based on these classes, Fig. S5i shows that in the promoter analysis, the correlation between dPC1 and RNA-seq again was the strongest when dPC1 and dPC2 had opposite signs. Thus, different ways of incorporating the absolute binding information produced similar findings.

Functional interpretation and gene set enrichment

In our examples, the biological meanings of dPCs were clear from looking at the patterns in $v_j$. In both the MYC and promoter analyses, dPC1 mainly represents differences driven by gene activation marks, and dPC2 mainly represents differences driven by the gene repression mark H3K27me3. In the ASB analysis, dPC1 reflects the correlated allele-specificity of a number of proteins associated with gene activation. In future applications, the biological meaning of the reported dPCs may not always be clear to users just looking at the $v_j$ pattern. In that situation, one may use other existing ‘omics data to help with interpreting the dPCs. Below we illustrate this using analysis of gene set enrichment.
We take the promoter analysis as an example. 3272 curated gene sets were downloaded from the MSigDB which is the gene set database used to support the Gene Set Enrichment Analysis. From these gene sets, we removed genes that were not associated with any promoters used in this article (association was determined by matching ENTREZ Gene Identifiers (IDs)). The remaining gene sets were used in the subsequent analyses. For each dPC, we grouped the analyzed promoters into three classes: (1) “+”, in which loci had significantly stronger ChIP-seq signals in K562 compared to Huvec; (2) “−”, in which loci had stronger signals in Huvec compared to K562; (3) “o”, which contained non-differential loci. Here we tried to define the classes using five different approaches, A, P, PR0.5, PR0.75 and Z, using the procedure described in the previous section. For instance, for PR0.5, “+” was defined by FDR ≤ 0.05, β_k > 0 and R^2 > 0.5; “o” was defined by FDR > 0.05; loci with FDR ≤ 0.05 but R^2 ≤ 0.5 were treated as ambiguous and therefore did not belong to any class, etc. Using these classifications, we obtained differential loci for the following four dPC classes for each method: (1) dPC1+, (2) dPC1−, (3) dPC2+, and (4) dPC2−. We annotated promoters in each dPC class using ENTREZ ID and removed redundancy in the data (i.e., within each class, if an Entrez ID was found to be associated with more than one promoters, only one promoter was retained). Next, we asked which gene sets were enriched in each of the four classes above. To do so, we obtained the following numbers: (1) N, the total number of non-redundant genes in MSigDB (only genes used in our annotations count), (2) n_+, the number of non-redundant genes in each dPC class c, (3) K_c, the number of genes in each gene set s, and (4) k_sc, the number of genes in gene set s that were found in dPC class c. For each dPC class c and gene set s, a Fisher’s exact test was then performed using the four numbers (N, n_+, K_c, k_sc) to test whether k_sc is significantly larger than what would be expected by chance. The p-values were converted into q-values to account for multiple testing. It turned out that different approaches for defining dPC classes produced similar results. Below we will use dPCA-P (P) to illustrate the findings.

The bar plot in Fig. S5j shows the top enriched gene sets identified for each dPC class defined by dPCA-P. For each gene set s, the total number of genes (i.e., K_s) and the number of genes found in class c (i.e., k_sc) are displayed in the brackets as (“k_sc/K_s”) next to the name of the gene set. The log2 fold enrichment, defined as r_sc = \log_2([k_sc + 1]/[K_s + 1])/[(n_+ + 1)/(N + 1)]) is shown as a bar. The significance level is indicated by stars (q-value ≤ 0.1: *; ≤ 0.01: **; ≤ 0.001: ***).

K562 is a cancer cell line established from chronic myeloid leukemia (CML). Huvec is established from normal umbilical vein endothelial cells. Gene sets associated with dPC1 were indicative of characteristics of these cell types (Fig. S5j). For example, for dPC1+ where gene activation chromatin marks are up in K562 compared to Huvec, we found enrichment of “VALK AML CLUSTER 7”, a gene set expressed in acute myeloid leukemia, consistent with K562 being a leukemia cell line. In contrast, among the top gene sets associated with dPC1− where the gene activation marks are stronger in Huvec, we found “UZONYI RESPONSE TO LEUKOTRIENE AND THROMBIN”, a gene set derived from studying genes expressed in Huvec cells, consistent with its association with Huvec. For dPC1−, we also found “GRAHAM CML QUIESCENT VS NORMAL QUIESCENT DN”, which contains genes down-regulated in quiescent (G0) CD34+ cells isolated from peripheral blood of CML (chronic myeloid leukemia) patients compared to the quiescent cells from normal donors. This is consistent with the lower level of gene activation marks in K562 at the loci in the dPC1− class. In addition, we also observed enrichment of “MANALO HYPOXIA DN” in dPC1+ and “MANALO HYPOXIA UP” in dPC1−. These two gene sets contain genes down- and up-regulated in response to both hypoxia and overexpression of an active form of HIF1A in arterial endothelial cells. Of note, Huvec also has an endothelial cell origin. Overall, these analyses suggest that gene sets expected to have a higher level of expression in one cell type were associated with higher dPC ChIP-seq signal in that cell type, consistent with dPC1 being a gene activation module.

As we discussed before, dPC2 mainly represents differences in H3K27me3. Interestingly, among the top gene sets enriched in dPC2- where H3K27me3 is up in Huvec compared to K562, we immediately found “BENPORATH ES WITH H3K27ME3” and “KONDO PROSTATE CANCER HCP WITH H3K27ME3”. Both these gene sets contain target genes bound by H3K27me3 in different cell types. Furthermore, we also found “BENPORATH SUZ12 TARGETS”, “BENPORATH EED TARGET” and “BENPORATH PRC2 TARGETS”, which contain target genes of SUZ12, EED and PRC2. SUZ12 and EED are components of Polycomb Repressive Complex 2 (PRC2), and PRC2 is involved in the trimethylation of H3K27 and gene repression [17]. This is again consistent with H3K27me3 being the major player in dPC2 and links dPC2 to gene repression. Importantly, if one had no prior knowledge about the biological meaning of dPC2 or about the mechanisms that modulate changes of H3K27me3, this analysis would allow one to connect dPC2 and H3K27me3 to SUZ12, EED and PRC2, and subsequently pointing to a potential new discovery. Among the top gene sets associated with dPC2+ where H3K27me3 is up in K562, we again found “UZONYI RESPONSE TO LEUKOTRIENE AND THROMBIN” and “MANALO HYPOXIA UP”, consistent with their expected lower expression in K562.

Together, our analyses suggest that in the future applications of dPCA, if the biological meaning of a dPC is not immediately clear by looking at the pattern in \( v_j \), one may use other existing ‘omic data such as pre-compiled gene sets to help with interpreting the biological functions of the dPC.

Using the dPC classes defined by A, PR0.5, PR0.75 and Z to perform the analysis produced similar results. For instance, for each gene set in Fig. S5j, Fig. S5k shows its log_2 FCs (i.e., r_sc) in the corresponding dPC class defined by the other four methods. All log_2 FCs are depicted positive. Fig. S5l shows that all these enrichments passed the q-value ≤ 0.1 cutoff.

In addition to analyzing dPCs separately, one may also analyze dPC1 and dPC2 jointly by applying the Fisher’s test to the nine (dPC1, dPC2) combinatorial patterns. The nine dPC classes can be defined using the procedure described in the previous section. Performing the joint analysis using different methods produced similar results. In the interest of space, below we only use two representative methods, P and PR0.5, to illustrate the main findings. Fig. S5m shows the top gene sets significantly enriched in each dPC class. For each method (i.e., P or PR0.5), two heat maps are shown. The left heat map shows the log2 fold enrichment (i.e., the \( r_sc \) defined above) of gene set \( s \) in dPC class c. The heat map on the right shows whether the enrichment of gene set \( s \) in class c was statistically significant (i.e., q-value ≤ 0.1). The results from P and PR0.5 were similar, and both were largely consistent with the results from analyzing each dPC separately. For instance, in the joint analysis, “UZONYI RESPONSE TO LEUKOTRIENE AND THROMBIN” was found to be associated with dPC1-dPC2+, consistent with its association with dPC1 reported by the separate analysis. The joint analysis also identified “BENPORATH ES WITH
H3K27ME3", "BENPORATH EED TARGETS" and "BENPORATH PRC2 TARGETS" to be associated with dPC2.

The joint analysis also provided new information. For instance, "BENPORATH PRC2 TARGETS" was found to be associated with all three subclasses of dPC2: dPC1+dPC2-, dPC1odPC2-, and dPC1-dPC2- in the joint analysis. In contrast, whereas "GUENTHER GROWTH SPHERICAL VS ADHERENT DN" was found to be associated with dPC1- in the separate analysis, the joint analysis shows that it was only significantly enriched in dPC1-dPC2+, but not in dPC1-dPC2-. Also, it is more enriched in dPC1-dPC2o than in dPC1-dPC2-. In contrast, "UZONYI RESPONSE TO LEUKOTRIENE AND THROMBIN" is also enriched in dPC1. However, the enrichment level of this gene set was higher in dPC1-dPC2- compared to dPC1-dPC2o. These examples show how users may characterize dPC functions in more detail by looking at dPCs jointly.

Fig. S2h-k shows the similar analysis for MYC. Here, we first annotated each motif site with its closest gene using CisGenome with its default parameter setting. We then used these annotations to convert the differential loci classes into gene classes. After removing redundancy within each class, we then performed the enrichment analysis. The analysis produced similar results. For instance, Fig. S2h shows a few top enriched gene sets obtained from dPCA-P. dPC1+ was again found to be associated with "VALK MAL CLUSTER 7", a leukemia gene set. dPC1- was associated with "UZONYI RESPONSE TO LEUKOTRIENE AND THROMBIN", a gene set derived from studying Huvec cells, and "GRAHAM CML QUIESCENT VS NORMAL QUIESCENT DN", a gene set down-regulated in CML leukemia. For dPC2+, we found "REACTOME NRAGE SIGNALS DEATH THROUGH JNK", consisting of genes involved in NRAGE cell death signaling through JNK. This gene set is involved in apoptosis which is usually expected to be inhibited in cancers. For dPC2-, we again found "BENPORATH ES WITH H3K27ME3" and "KONDO PROSTATE CANCER HCP WITH H3K27ME3", which links dPC2 to H3K27me3 and PRC2. Enrichment analyses using dPC classes defined by A, PR0.5, PR0.75 and Z gave similar results (Fig. S2i,j). Similar to the promoter analysis, analyzing dPC1 and dPC2 jointly can provide some new information compared to analyzing dPCs separately (Fig. S2k). For instance, the joint analysis again found that "BENPORATH PRC2 TARGETS" was enriched in all three subclasses of dPC2: dPC1+dPC2-, dPC1odPC2-, and dPC1-dPC2-. In contrast, "GRAHAM CML QUIESCENT VS NORMAL QUIESCENT DN" was found in dPC1- in the separate analysis. The joint analysis shows that it was only significantly enriched in dPC1-dPC2- but not in dPC1-dPC2o.

Fig. S6l-n shows the enriched gene sets associated with dPC1 in the ASB analysis. Here, SNPs were annotated with their closest genes using CisGenome. We did not break dPC1 into dPC1+ and dPC1-, since the reference and non-reference alleles were randomly determined in the sequencing project and are not expected to be different in terms of their biological functions. Again, A, P, PR0.5, PR0.75 and Z produced similar results. All methods found that gene sets enriched in dPC1 include those related to immune system (e.g., "REACTOME SIGNALING IN IMMUNE SYSTEM", "KEGG ANTIGEN PROCESSING AND PRESENTATION") and cancer metastasis (e.g., "BIDUS METASTASIS DN").

Discussion
Absolute binding. In dPCA, multiple options are provided to incorporate absolute binding into the analysis. Differences not associated with significant binding are not easy to interpret. They could represent differences in non-specific binding or backgrounds of the two conditions not captured by the cross-replicate variation. In this case, one may want to filter them out. However, they could also represent subtle binding but real biological differences, and such differences may be of interest to an investigator. For instance, when one studies the initiation of a regulatory program during development, one may want to look at the dynamic process in which the binding level progressively changes from no binding to low binding and then to strong binding. Changes between two nearby time points during the initiation phase of the process may occur at loci without significant binding, but may provide important information regarding when the program is initiated. In this case, filtering out loci without significant binding may filter out important biological information. Thus, there may not be a simple answer regarding whether one should include or exclude differences not associated with significant binding in the analysis. For these reasons, we designed dPCA in a modular fashion so that on top of the basic model which does not use the absolute binding information, we have multiple options for incorporating absolute binding into the analysis. Users can choose whether or not to use them and how to use them based on the needs of their specific projects. In this article, we chose to filter out loci without any significant binding activity before the analysis to ensure that our findings are not a result of analyzing differences in non-specific binding or backgrounds.

In all our examples, handling absolute binding in different ways produced similar dPC patterns but varying numbers of loci. Evaluation based on different types of gold standard (i.e., MYC binding, RNA-seq, chrX) show that dPCA-P (or AP) often performed well both in terms of the ranking performance and the overall sensitivity for detecting differences in the gold standard data (note: differences not covered by these gold standards can be real but we do not have enough information to tell and to benchmark them). Applying the $R^B$ filter after dPCA-P allows one to more stringently identify loci whose main differences and binding occur in the same datasets. This may be useful if users want to identify a cleaner set of differential loci with strong binding for which interpretation and follow-up experiments could be easier. On the other hand, increasing the stringency may result in losing sensitivity and does not necessarily improve the overall ranking. Thus if the goal is to more comprehensively discover differential loci or predict signals such as differential TF binding or RNA-seq, or if one has other sources of information (e.g., results from functional studies or other screens) that may be used to remove false positives after the dPCA analysis, users may choose not to apply the $R^B$ filter. dPCA-Z removes differences not associated with significant binding before dPCA. It may help one to verify that the patterns discovered by dPCA are not dominated by differences in background or non-specific binding. In our examples, the stringency level of dPCA-Z was between $R^B > 0.5$ and $R^B > 0.75$ (Fig. S7j,k). An advantage of using $R^B$ over dPCA-Z is that $R^B$ is reported for each locus and dPC, therefore it is more convenient for users to adjust the stringency of the filter after they obtain the dPCA results. In our data, $R^B > 0.5$ offered a good balance if one wants to both identify differential loci with significant amount of binding (Fig. S7n-d) and keep the ranking performance competitive (Fig. S7f-g), therefore it is often used by us if we use $R^B$. However, just as using $p$-values to define significance, users can adjust the $R^B$ cutoff depending on how stringent they want to be.
Integrative analysis of quantitative differences in multiple ChIP-seq datasets is a problem not extensively studied previously. Analyzing differences and absolute binding together is even more challenging. While we provided some initial solutions to this problem, they may not be the only or the final solution to this problem. As Fig. S7h shows, the same amount of differences at loci with different absolute binding levels may have different probabilities to be associated with differential gold standard signals. This suggests that there might be rooms for improving the ranking and statistical tests in our current method by better integrating the absolute binding information into the analysis of differences. However, how this improvement can be implemented remains unknown and is not necessarily trivial. There could be many possible ways to extend dPCA to directly integrate absolute binding into the model. For instance, one may assume that the prior probability for being differential along dPCA j is a locus- and dataset-specific prior $\pi_j$ that depends on absolute binding, rather than being $\mathbb{E}_j$ which is equal across loci. Instead of assuming common variance $\sigma^2$, one may also assume that errors have locus- and dataset-specific variances $\sigma^2_m$ and then model the dependency between $\sigma^2_m$ and the absolute binding $\mu_m$ at locus $g$ in dataset $m$. This may be further extended by modeling $\sigma^2_m$ as a function of $\mu_m$ and their interactions from all datasets. Besides developing these possible model extensions, one may also improve our current approach of using absolute binding as a filter by developing better ways to summarize the absolute binding information. For instance, $R^B$ may be generalized by using $\gamma \mu_m \sigma^2_m$ to replace $\mu_m \sigma^2_m$, where $\gamma$ is a number whose optimal value needs to be explored. One may also develop a statistic similar to $R^B$ but not dependent on peak calling cutoffs. Investigating these topics is part of our ongoing research. There could be many other possibilities not listed here to generalize dPCA. Enumerating and exploring all these possible extensions is beyond the scope and capacity of the current paper. Users of dPCA are encouraged to routinely check the dPCA website for updates and technical reports discussing possible extensions and new developments of dPCA. In summary, looking for better ways to analyze differences by integrating information from absolute binding is still an open problem. It is our hope that the work here can stimulate more research in this area in the future.

The zero-mean assumption. dPCA assumes $E(\delta_g) = 0$ which implies that $d_{gm}s$ in each dataset are bidirectional and fluctuate around zero. This assumption is reasonable in a wide range of applications including our examples (see Fig. S1d,e). When it is not reasonable to make this assumption, e.g., if $d_{gm}s$ in some datasets are unidirectional and all positive, one may run dPCA after centering column means to zero, but the results need to be interpreted differently: the dPCs and $\delta_g$s will characterize variations around the global mean of the differences across all loci, rather than variations around zero; the reported loci will be those whose $\delta_g$s are significantly different from an average locus, rather than different from zero. In certain applications, a unidirectional problem may be converted into a bidirectional problem to allow one to characterize changes around zero. For example, to predict differential TF binding from differential chromatin patterns, unidirectional binding changes of a TF may be inferred by analyzing motif sites of many TFs together. While some TFs change in one direction, other TFs may change in an opposite direction. Thus when all motif sites are pooled together, $d_{gm}s$ of HMs will be bidirectional and fluctuate around zero.

Equal variance versus locus- and dataset-specific variances. In dPCA, the replicate variance $\sigma^2$ is assumed to be equal for different loci and datasets. Even though this assumption may not hold true in real data, our simulations with real data characteristics show that the dPC and FDR estimates are not severely biased as long as the SNR is big, suggesting that this assumption is a reasonable one to use. We also explored the possibility to modify the algorithm to allow locus- and dataset-specific variances. In the modified model, we assume that elements in $\mathbf{E}$ are independent Gaussian random variables $\epsilon_{gm} \sim N(0, \sigma^2_m/(1/K_{1m} + 1/K_{2m}))$ (i.e., $\sigma^2$ is replaced by $\sigma^2_m$). Under this new assumption, $\text{Var}(\delta_g) = E(\delta_g^2) = E(\text{d}_g^2 - \text{e}_g^2) = E(\text{d}_g^2) - E(\text{e}_g^2) = E(\text{d}_g^2) - \Omega_g$, where $\Omega_g = \text{diag}(1/K_{1g} + 1/K_{2g})\sigma^2_{1g}$, $\ldots, (1/K_{1M} + 1/K_{2M})\sigma^2_{Mg}$. Thus, $E(\Delta G) = \text{Var}(\delta_g) = \sum_g E(\text{d}_g)^2 / G - \sum_g \Omega_g / G = E(D^T \text{D}) / G - \sum_g \Omega_g / G$. $E(D^TD/G)$ can be estimated using $D^T \Sigma D/G$. To estimate $\Omega_g$, one only needs to estimate $\sigma^2_m$. To do so, we used an empirical Bayes shrinkage estimator described in [24]. Briefly, we first compute the sample variances $\hat{\sigma}^2_g$ for each locus $g$ and dataset $m$: $\hat{\sigma}^2_g = \sum_{k}(x_{gmk} - \bar{x}_{gm})^2 / \eta_m$ where $\eta_m = \sum_y (K_{ym} - 1)$. For each dataset $m$, we then compute $\hat{\sigma}^2_m = \sum_g \hat{\sigma}^2_g / G$ and $\hat{\sigma}^2_{gm} = \sum_g \hat{\sigma}^2_g - \hat{\sigma}^2_m$. The $\hat{\sigma}^2_{gm}$ is then estimated using $\hat{\sigma}^2_{gm} = (1 - B_m)\hat{\sigma}^2_m + B_m \hat{\sigma}^2_g$. Here, $B_m$ takes a value between 0 and 1. It is computed as:

$$B_m = \min \left(1, \frac{2(G - 1)}{(2 + \eta_m)G} + \frac{2(G - 1)\hat{\sigma}^2_m}{(2 + \eta_m) \sum_g \hat{\sigma}^2_g} \right)$$

$B_m$ is a dataset-specific shrinkage factor that pulls the sample variances $\hat{\sigma}^2_g$s toward their global mean to produce the estimates for $\sigma^2_{gm}$. As shown in [24], this shrinkage estimator can achieve better bias-variance tradeoff and makes the variance estimates more stable when the number of replicates is small. If a dataset $m'$ has only one sample in each condition, then we will have no degree of freedom to estimate $\sigma^2_{gm'}$. In that scenario, we take advantage of the replicate samples from the other datasets and use the pooled sample variances $\hat{\sigma}^2_g = \sum_g \sum_y \sum_k(x_{gmk} - \bar{x}_{gm})^2 / \sum_y (K_{ym} - 1)$ to serve as $\hat{\sigma}^2_{gm}'$ and use $\sum_g \sum_y (K_{ym} - 1)$ to serve as $\eta_{gm}'$ for that dataset.

After obtaining $\hat{\sigma}^2_{gm}$, we plug them into $\Omega_g$, and estimate $E(\Delta G)$ by $D^T \Sigma D / G - \sum_g \Omega_g / G$. We then use the eigen decomposition of the estimated $E(\Delta G)$ to estimate $\mathbf{V}$. Projecting the data to each dPCA yields $\mathbf{v}_g^T \mathbf{d}_g = \mathbf{v}_g^T \delta_g + \mathbf{v}_g^T \mathbf{e}_g = \beta_g + \epsilon_g$, where $\epsilon_g \sim N(0, \rho \mathbf{v}_g \mathbf{v}_g^T)$. Accordingly, one can construct the t-statistic to detect and rank genic loci along each dPCA: $T_g = \mathbf{v}_g^T \mathbf{d}_g / \sqrt{\hat{\sigma}^2_g \mathbf{v}_g \mathbf{v}_g^T}$. For the paired sample situation (e.g., ASB), the above procedure can be applied with slight modification. First, define $y_{gmk} = x_{gmk} - x_{g2mk}$ and note that $K_{1m} = K_{2m} = \hat{K}_m$. If a dataset $m$ has multiple replicates, then $\hat{\sigma}^2_{gm} = \sum_k(y_{gmk} - \bar{y}_{gm})^2 / \eta_m$ where $\eta_m = \hat{K}_m - 1$. If a dataset $m'$ has no replicate samples, then we use $\hat{\sigma}^2_{gm} = \sum_k(y_{gmk} - \bar{y}_{gm})^2 / \sum_y (K_{ym} - 1)$ to serve as $\hat{\sigma}^2_{gm}'$, and use $\sum_y (K_{ym} - 1)$ to serve as $\eta_{gm}'$ for that dataset. $\Omega_g = \text{diag}(\sigma^2_{1g}/\hat{K}_1, \ldots, \sigma^2_{Mg}/\hat{K}_M)$ can be estimated by plugging in the estimates of $\sigma^2_{gm}'$.

Instead of using the empirical Bayes variance shrinkage estimator, one can also directly use $\hat{\sigma}^2_g$ to estimate $\sigma^2_{gm}'$ in the above algorithms. We applied the modified dPCA both with and without variance shrinking to real data. Fig. S4d shows ranking analyses similar to Figs. 2c,4d,5c,d. The modified dPCA did not bring obvious improvement over the original dPCA based on the equal variance assumption.
Instead of using locus- and dataset-specific variances, we also tested using locus-specific variances by assuming \( \epsilon_m \sim N(0, \sigma_m^2(1/K_{1m} + 1/K_{2m})) \). This is a special case of the locus- and dataset-specific variance model with the additional assumption \( \sigma^2_{ij} = \ldots = \sigma^2_{jm} \). Under this assumption, one can similarly use an empirical Bayes shrinkage estimator to estimate \( \sigma^2_m \). Briefly, one first computes sample variance \( \tilde{\sigma}^2_m = \sum_i \sum_j (x_{ijm} - \bar{x}_{jm})^2 / \eta_m \) where \( \eta_m = \sum_i \sum_j (K_{jm} - 1) \). One then computes \( \overline{\sigma}^2 = \sum \overline{\sigma}^2 / G \) and \( S = \sum (\overline{\sigma}^2 - \overline{\sigma}^2)^2 \). The \( \overline{\sigma}^2 \) is estimated as \( \overline{\sigma}^2 = (1 - B\overline{\sigma}^2) + B\sigma^2 \), where \( B \) is calculated using Eq. 2 after replacing \( \eta_m \) by \( \eta_m \) replacing \( \overline{\sigma}^2 \) by \( \overline{\sigma}^2 \), and replacing \( S_m \) by \( S \). Similarly, with slight modification, this approach can also be applied to the paired sample scenario.

Finally, we also tested dataset-specific variances by assuming \( \epsilon_m \sim N(0, \sigma^2_m(1/K_{1m} + 1/K_{2m})) \). This is a special case of the locus- and dataset-specific variance model with the additional assumption \( \sigma^2_{im} = \ldots = \sigma^2_{jm} \). Under this assumption, we estimate \( \sigma^2_m \) by \( \tilde{\sigma}^2_m = \sum_i \sum_j (x_{ijm} - \bar{x}_{jm})^2 / \eta_m \), where \( \eta_m = G \times \sum_j (K_{jm} - 1) \). Here variance shrinking is not necessary since one has large enough degrees of freedom (because of large \( G \)) to stably estimate \( \sigma^2_m \). If a dataset \( m' \) does not have replicate samples, we use \( \sigma^2 = \sum_i \sum_m \sum_j (x_{ijm} - \bar{x}_{jm})^2 / \eta_m \) to serve as \( \tilde{\sigma}^2_m \) where \( \eta = G \times \sum_j (K_{jm} - 1) \) and we use \( \eta \) to serve as \( \eta_m' \) for that dataset. With \( \tilde{\sigma}^2_m \) and \( \overline{\sigma}^2 \) available, the other parts of dPCA remain the same. Paired sample cases can be handled similarly with slight modification.

Based on real data analyses in Fig. S4d, using locus-specific variance or dataset-specific variance, regardless of whether the variance shrinkage estimator was used or not, still did not improve the performance over the original equal variance dPCA. In fact, the original dPCA performed the best in the MYC data and among the best in the Promoter and ASB data for ranking differential loci. Based on these analyses, we decided to use the original dPCA based on the equal variance assumption in our software implementation. Note that the variance models tested above did not use the absolute binding information. As we have discussed before, whether or not directly integrating absolute binding information into the model can improve the dPCA performance is still an open problem that requires future investigation.

### Why are the dPCs with small eigenvalues ignored?

When there are multiple datasets, analyzing differences between two conditions is more than simply report which loci are differential. One often also wants to find patterns of differences and organize differential loci based on the patterns. To discover the covariation patterns across multiple datasets, dPCA attempts to study the variance structure of the unknown true differences \( \Delta \). To do so, it removes the variance of the error components \( \epsilon \) from the observed data variance. With the error components removed, all elements in \( \Delta \) are viewed as true between-dataset differences that cannot be explained by the cross-sample variability. Correspondingly, all principal components of \( \Delta \) explain certain fraction of the true differences.

A natural question is why we only report a few leading dPCs and ignore the others if they are all real differences. Besides considerations of dimension reduction, there is an important reason for this: if pattern discovery is a goal of the analysis, it is important that the patterns reported by dPCA can accurately reflect the underlying truth and are also reproducible. In other words, ideally one would like to have \( \psi_j \) close to the true \( \psi_j \). Moreover, if other investigators plan to generate similar ChIP-seq data again in the future and apply dPCA to their new data, they should be able to find roughly the same patterns. Being able to reproduce the discoveries is important for scientific research. If the reported patterns are not reproducible, it is not meaningful to spend time and money to perform detailed follow-up studies as no other labs will be able to verify the findings.

Unfortunately, with the finite number of genomic loci \( G \) (even though \( G \) is large), dPCs with small eigenvalues often have small signal-to-noise ratio and cannot be estimated accurately and discovered reproducibly. Figs. 3 and S3 show that as SNR decreases, the distance between \( \psi_j \) and the true pattern \( \psi \), increases, and dPCs with small SNR usually deviate significantly from the underlying true \( \psi \). Moreover, for dPCs with small SNR, the \( \hat{\psi}_j \) estimates are also highly variable, as indicated by the wide error bars in Figs. 3b, S3(2-h2) and the simulations in Fig. S4a-c. In Fig. S4a-c, we performed simulations to mimic the situation in which two hypothetical labs independently generate similar data and then analyze their data using dPCA. For each example – MYC, promoter and ASB – a pair of simulations was performed. The simulation data were created using the same procedure used to generate Figs. 3 and S3 (under the \( \tau_0 = 0.25 \) setting) to capture real data characteristics. Data within each pair of simulations were generated from the same underlying true model. Fig. S4a-c shows that the distance between the \( \psi_j \) from the two simulations increased as we moved from leading dPCs to smaller dPCs with smaller SNR (see SNRs in Figs. 3, S3). For leading dPCs, the \( \psi_j \) from the two simulations were consistent. For the small dPC components, the two hypothetical labs discovered very different patterns, as evidenced by the large distance between their \( \psi_j \). This shows that the estimated differential patterns for dPCs with small eigenvalues and SNRs are unreliable and cannot be reproduced if independent labs were to generate similar data and apply the same analysis.

Intuitively, each dPC represents a geometric direction in the \( \mathbb{R}^G \) space. When the SNR for a dPC is small, the dPC estimate \( \psi_j \) can be easily rotated in the geometric space due to high level of noise to point to a direction very different from the true \( \psi_j \). Therefore, dPCs reported by analyzing two similar but independent data collections may significantly deviate from each other. When the pattern estimate \( \hat{\psi}_j \) is biased, data projections to \( \hat{\psi}_j \) will be very different from data projections to the true \( \psi_j \). Consequently, statistical inference for \( \beta_{\psi_j} \) such as the FDR estimates will also be biased and deviate from the truth. For these reasons, although dPCs with small eigenvalues and small SNRs represent true differences, one cannot accurately and reproducibly recover them. This is analogous to a situation where one knows that there exist many true positives (e.g., truly differentially expressed genes in microarray data) hidden in the noisy data but cannot identify all of them because of the noise. This explains why the small dPCs are not reported by dPCA in its default mode.

Naturally, the next question is how one can determine which dPCs to report. Empirically, our simulation studies in Figs. 3 and S3 showed that \( SNR > 5 \) was a cutoff that produced reasonably accurate dPC estimates as well as reasonable FDR estimates in a wide range of parameter settings. For this reason, dPCA uses \( SNR > 5 \) as its default cutoff to determine which dPCs to report. This explains why two dPCs were discussed in the MYC and promoter analyses, whereas only one dPC was discussed in the ASB analysis.

In our software, users have the option to change the SNR cutoff. If pattern discovery is not the primary goal and users just want to identify all differential loci with any differences, they could lower the cutoff to allow dPCA to report all dPCs instead of a few top dPCs. A locus reported by any dPC can
Detecting common differences versus detecting unique differences. dPCA attempts to find major differential patterns in the data, such as common patterns shared by many loci. If there are no such patterns, one may not be able to find any significant dPC directions. In that scenario, describing the differences based on dPCs may not be useful. Also, for $M$ datasets, there will be $3^M$ possible combinatorial patterns of differential status (up, down, no change). Even if there are some major patterns shared by many loci, it is possible that certain genomic loci have relatively unique patterns that are different from all other loci. When investigators want to study these loci with relatively unique patterns, dPCA may not help directly. In that case, one may need to use other methods to characterize differential PDL. For example, one simple approach is to identify differential loci by performing a t-test for each locus in each dataset. The tests will give $p$-values which can then be converted to q-values or FDR to account for multiplicity. For each locus, one can compute the minimal q-value across all datasets and use it to indicate whether the locus is differential in any dataset. Genomic loci can then be ranked based on this minimal q-value statistic to identify top differential loci. Loci identified in this way but not identified by dPCA will likely be interesting for those who want to study loci with unique differential patterns.

How to apply dPCA when no replicate sample is available. Replicate samples carry important information for examining technical reproducibility and biological variability. For this reason, people increasingly include replicate samples in Chip-seq studies. Many important projects such as the ENCODE now require investigators to generate replicates to ensure data quality. Consistent with this trend, dPCA is primarily designed for analyzing data with replicate samples. Under the equal variance assumption, as long as there is one dataset that contains two or more replicates, one has degree of freedom for estimating the variance $\sigma^2$. Thus, dPCA can be applied even if some of the datasets do not have replicate samples.

For users who do not have any replicate in their data, a modified version of dPCA is provided so that the data can still be analyzed. Below we describe this variant of dPCA. However, we would like to point out that analyzing data without replicates requires very different new model assumptions, and a comprehensive treatment of this topic requires an article-long discussion which is beyond the scope of the present paper. The solution below is only provided for users’ convenience and is not meant to be optimal.

First, consider the non-paired sample case (e.g., data from Examples I and II). Since there is no replicate, we drop the subscript $k$ and use $x_{gim}$ to denote the binding intensity for locus $g$, condition $i$ and dataset $m$. Assume that each $x_{gim}$ has variance $\sigma^2$ reflecting its cross-sample variability (even though it may not be observed since no replicate sample is available). Without replicates, there is no degree of freedom for estimating $\sigma^2$ unless one is willing to make additional assumptions. The basic idea behind our solution below is that if one can obtain a list of non-differential loci in each dataset, one may treat the data from two biological conditions, $x_{g1m}$ and $x_{g2m}$, at those loci as two replicate samples. This will provide the degrees of freedom for estimating $\sigma^2$. Once $\sigma^2$ is estimated, all the other parts of dPCA will remain the same.

In order to obtain a putative list of non-differential loci, we use the following approach. Define $d_{gm} = x_{g1m} - x_{g2m}$. Assume that different loci are independent. Within each dataset $m$, $d_{gm}$ are assumed to follow a mixture distribution:

$$d \sim q_{0m} f_{0m}(d) + q_{1m} f_{1m}(d)$$

Here $f_{0m}(\cdot)$ and $f_{1m}(\cdot)$ are probability density functions used to describe $d_{gm}$s from non-differential and differential loci respectively. $q_{km} (k = 0, 1)$ are prior probabilities for these two components, and $q_{0m} + q_{1m} = 1$. In our implementation, a normal distribution $N(0, \sigma^2_m)$ with unknown variance $\sigma^2_m$ is used to serve as $f_{0m}(\cdot)$, $f_{1m}(\cdot)$ is assumed to be an uniform distribution $U[b_{0m}, b_{1m}]$, where $b_{0m}$ and $b_{1m}$ are treated as known parameters and are set to be the minimum and maximum of $d_{gm}$s across all loci in dataset $m$.

Under this model, an Expectation-Maximization (EM) algorithm [25] can be derived to estimate the parameters $q_{km}$ and $\sigma^2_m$. Briefly, let $a_{gkm} = 1$ or 0 indicate whether locus $g$ is differential in dataset $m$ or not, $a_{gkm} = 1 - a_{gkm}$. $q_{km} = \{q_{0km}, q_{1km}\}$, $A_m = \{a_{gkm} = 1, \ldots, G; k = 0, 1\}$, and $D_m = \{d_{gm} : g = 1, \ldots, G\}$. Using these notations, the complete data likelihood for dataset $m$ is

$$Pr(D_m, A_m|q_m, \omega^2_m) = \prod_{g=1}^G \prod_{k=0}^1 \left( q_{0km} f_{0km}(d_{gm}) \right)^{a_{gkm}}$$

The complete data log-likelihood, $\log Pr(D_m, A_m|q_m, \omega^2_m)$, is

$$l(q_m, \omega^2_m, D_m, A_m) = \sum_k \left( \sum_g q_{gkm} \log q_{gkm} - \sum_g q_{gkm} \frac{d_{gm}^2}{2\sigma^2_m} \right)$$

$C$ is a constant that does not involve unknown parameters. EM is an iterative algorithm. Let $q_{km}^{(t)}$ and $\omega^2_m^{(t)}$ be the estimates of $q_{km}$ and $\omega^2_m$ at iteration $t$. The algorithm will iteratively update these estimates through an E-step and an M-step. In the E-step, the expectation of $l(q_m, \omega^2_m, D_m, A_m)$ is evaluated with respect to the conditional distribution of $a_{gkm}$ given $D_m$, $q_m$ and $\omega^2_m$. This produces a Q-function:

$$Q(q_m, \omega^2_m|q_m^{(t)}, \omega^2_m^{(t)}) = \sum_k \left( \sum_g q_{gkm}^{(t)} \log q_{gkm} - \sum_g q_{gkm}^{(t)} \frac{d_{gm}^2}{2\sigma^2_m} \right) + C$$

where

$$q_{gkm}^{(t)} = Pr(a_{gkm} = 1|D_m, q_m^{(t)}, \omega^2_m^{(t)}) = \frac{q_{gkm}^{(t)} f_{a_{gkm}}^1(d_{gm})}{\sum_k q_{gkm}^{(t)} f_{a_{gkm}}^1(d_{gm})}$$

Here, $f_{a_{gkm}}^0(\cdot)$ is the density for $N(0, \omega^2_m^{(t)})$, and $f_{a_{gkm}}^1(\cdot)$ remains to be the uniform distribution $U[b_{0m}, b_{1m}]$.

In the M-step, the algorithm searches for $q_m$ and $\omega^2_m$ that maximize the Q-function. This gives parameter updates:

$$q_{gkm}^{(t+1)} = \frac{\hat{a}_{gkm}^{(t)}}{G}$$

$$\omega^2_m^{(t+1)} = \sum_g \frac{d_{gm}^2}{2}$$
The new parameter estimates are then used to carry out the next EM iteration. Upon convergence, the algorithm will give an estimate (which can be a local mode) that maximizes the observed data likelihood $P(D|m, \sigma^2, \omega_m)$ in which the missing data $A_m$ are integrated out. Using the estimated parameters $\hat{q}_m, k$ and $\hat{\omega}_m^2$, one can compute the posterior probability that locus $g$ is not differential in dataset $m$:

$$q_{gm0} = \frac{\hat{q}_{m0} \hat{f}_{m0}(d_{gm})}{\hat{q}_{m0} \hat{f}_{m0}(d_{gm}) + q_{m1} \hat{f}_{m1}(d_{gm})}$$

(10)

Here $f_{m0}(\cdot)$ is the normal density $N(0, \hat{\omega}_m^2)$. We then collect all loci with $\hat{q}_{m0} > 0.9$ as the putative non-differential loci in dataset $m$. The set of such loci is denoted as $W_m$. Using the putative non-differential loci, we compute $\eta = \sum_m |W_m|$ where $|W_m|$ is the number of loci in the set $W_m$, and estimate $\sigma^2$ by

$$\hat{\sigma}^2 = \sum_m \sum_{i \in W_m} (x_{gim} - \hat{x}_{g1m} + \hat{x}_{g2m} \hat{\omega}_m^2)^2$$

(11)

With $\hat{\sigma}^2$ and $\eta$ available, one can now continue with dPCA as usual. In other words, one can first compute $D^T\hat{D}/\eta = \hat{\sigma}^2\Omega$. Here all $K_{gm}$ in $\Omega$ are equal to one. One can then use the eigendecomposition to estimate $\hat{V}$. Next, the data can be projected to each dPC to construct the t-statistics for detecting and prioritizing differential loci. We call this variant of dPCA “dPCA-nr”.

To illustrate dPCA-nr, we applied it to the data in Examples I (58997 MYC motif sites) and II (22368 promoters). In each example, to mimic a no-replicates situation, we randomly chose one sample from each dataset $m$ and condition $i$. Fig. S8a-c shows the results for the MYC analysis. Fig. S8a shows that the first two dPCs obtained from dPCA-nr were similar to those obtained by applying dPCA to the full data with all replicates (Fig. 2a). Comparing Fig. S8b and Fig. 2b, the SNR estimates from dPCA-nr were smaller compared to those obtained by applying dPCA. dPCA-nr will not report dPC2, whereas $\hat{\sigma}$ and FDR estimates were reasonable (e.g., dPC1 in Fig. S8b). When $5 < \text{SNR} < 10$, they were reasonable but started to show biases (e.g., dPC1 in Fig. S8b). When SNR $< 5$, both estimates can be very biased (e.g., dPC2 in Fig. S8b).

Next, we examined the effects of the new assumption introduced in the dPCA-nr, that is, data from two biological conditions at the putative non-differential loci can provide an estimate of the true cross-replicate variability $\sigma^2$ within each biological condition. To do so, we repeated the dPCA-nr analysis on the same simulation data without replicates, but now we replaced $\hat{\sigma}^2$ in dPCA-nr by an ideal estimate one would obtain from analyzing data with replicate samples. In other words, since we generated replicates in the simulations, we could use the replicate samples to estimate $\sigma^2$, just as estimating $\sigma^2$ in the full data dPCA. We could then use this $\hat{\sigma}^2$ to replace the $\hat{\sigma}^2$ in the dPCA-nr. Afterwards, we still apply dPCA-nr to the reduced data without replicate sample. The results from this analysis are shown in Fig. S8j,k by blue curves labeled as “dPCA(truevar)”. We note that in real applications, dPCA(truevar) cannot be applied if there is no replicate sample. This approach is compared here only for helping us to understand dPCA-nr which is represented by the black curves labeled as “dPCA(norep)”. Since both dPCA(norep) and dPCA(truevar) were applied to the same data without replicates, the $K_{gm}$ in both analyses were equal to one. Therefore the two analyses had the same $\Omega$. The only difference between them was the difference in the $\hat{\sigma}^2$ estimates. The simulation shows that compared to dPCA(truevar), dPCA(norep) (i.e., dPCA-nr) tends to overestimate $\sigma^2$, consistent with our observations in the real data. For instance, for the $\pi_0 = 0.5$ case, the $\hat{\sigma}^2$ estimates and their associated standard deviations produced by dPCA(truevar) and dPCA(norep) were 0.24 (SD = 0.029) and 0.48 (SD = 0.042) respectively. When $\pi_0 = 0.1$, the estimates were 0.23 (SD = 0.022) for dPCA(truevar) and 0.25 (SD = 0.030) for dPCA(norep). As a result, compared to dPCA(truevar), dPCA(norep) tends to have smaller SNR estimates (Fig. S8(j1,k1)) and more conservative FDR estimates (Fig. S8(j5-j7,k5-k7)) due to its overestimated $\hat{\sigma}^2$. Because of the smaller SNR estimates, dPCA(norep) sometimes reported
fewer dPCs than dPCA(truevar). For instance, in Fig. S8j, dPCA(truevar) reported two dPCs using the SNR > 5 cutoff, but dPCA(norep) only reported one dPC at the same cutoff. Since dPCA(norep) tends to underestimate the SNR compared to dPCA(truevar), one may think that reducing the SNR cutoff for dPCA(norep) may allow one to recover all dPCs one would obtain from dPCA(truevar). However, in our simulations, we did not find a consistent way to adjust the cutoff for dPCA(norep) in order to achieve this. For example, while reducing the cutoff in Fig. S8(j1) may allow dPCA(norep) to report the top two dPCs as one would like to have, decreasing the SNR cutoff in Fig. S8(k1) below 5 will result in unwanted patterns to be reported. In fact, in Fig. S8(k1), the SNR for dPC1 was slightly above 5, and its FDR estimates have already started to show some bias. Overall, since SNR > 5 remained to be a reasonable cutoff in our simulations, we retained it as the default cutoff for dPCA-nr.

Consistent with the conservativeness of dPCA(norep) in terms of estimating FDR, dPCA(norep) tends to report fewer differential loci at the same nominal FDR cutoff. Take the simulation in Fig. S8j as an example. At the 5% FDR level, dPCA(norep) and dPCA(truevar) reported an average of 5543 (SD = 162) and 7359 (SD = 624) differential loci for dPC1 respectively. For dPC2, they reported an average of 1977 (SD = 162) and 7359 (SD = 624) differential loci for dPC1 respectively. Now consider the paired sample scenario (e.g., ASB). Here the data to be analyzed are γgm = xgm - xgm, where xgm,m and xgm,2 may be correlated. We have K_m = 1 and dgm = ygm. Instead of imposing assumptions on xgm,m and xgm,2, we directly model ygm and assume that the cross-replicate variance of ygm is σ^2. Clearly, when there are no replicate samples, there is no degree of freedom for estimating σ^2. Even at the non-differential loci, each locus g only has one observation ygm in each dataset m. Therefore, the previous dPCA-nr approach which uses samples from two biological conditions as independent replicate samples to estimate σ^2 can no longer be applied here. In order to use dPCA, we make a new assumption that within each dataset m, ygm is from all non-differential loci can be viewed as replicates for each other. Under this assumption, the variability across the non-differential loci within a dataset can be used to estimate the variability across replicate samples within a locus.

We again use the mixture distribution in Eq. 3 to describe dgm,n in each dataset m. Now this is also the distribution for ygm. Based on our assumption, the variance ω^2 of the background normal distribution can provide an estimate for the cross-replicate variability σ^2. Therefore, after fitting the model by EM, one may use an average or weighted average of ω^2 to serve as σ^2. In light of Eq. 9, ω^2_m can be computed as follows. First, using the q_m and y_m produced by the last EM iteration and Eq. 10, one can compute q_m and y_m, the posterior probability that locus m is not differential in dataset m. Next, one can compute y_m = \sum_{g=1}^{G} q_m g_m and \hat{\omega}_m = \sum_{g=1}^{G} q_m g_m dgm/n based on Eq. 9. Thus approximately η_m degrees of freedom are used to estimate \hat{\omega}_m in dataset m. Therefore, we use the following weighted average σ^2 = \sum_m η_m \hat{\omega}_m / \sum_m η_m to serve as the final estimate for σ^2, and use η_m to estimate η. With these estimates available, the paired sample dPCA can be performed as usual. This algorithm will be called “dPCA-pnr” hereinafter.

We applied dPCA-pnr to the ASB data after randomly selecting one pair of samples from each dataset m to mimic a no-replicate situation. The results are shown in Fig. S8g-i. Figs. 5a and S8g show that dPC1 obtained from dPCA-pnr was similar to the dPC1 obtained from the full data dPCA. Fig. S8i shows that the SNP ranking based on dPC1 provided by dPCA-pnr remained to be the best compared to rankings based on individual datasets. However, Fig. S8j shows that in this example no dPC passed the SNR > 5 cutoff. Similar to our previous discussions, part of the reason was that K_m became smaller which increased Ω. Another reason was that σ^2 obtained by dPCA-pnr was also conservative. It was bigger than the σ^2 produced by applying dPCA to the full data with replicate samples (0.366 vs. 0.361). This is likely because the heterogeneity among different non-differential loci in real data is bigger than the heterogeneity among true replicate samples within each locus. It is also possible that the mixture model used here tends to be conservative in claiming which loci are differential, and therefore many differential loci may be called as non-differential loci which then make the background variance estimate bigger than its real value. Since no dPC passed the default SNR cutoff, in practice the dPCA-pnr will not report any dPC for this example.

We again studied the behavior of dPCA-pnr further using simulations (Fig. S8i-n). The data were simulated using the same procedure described in Example III. Again, the parameter π was used to control the overall SNR level. When π was 0.1, no dPC passed the SNR > 5 cutoff (Fig. S8m), similar to the real data. Therefore we displayed results from two additional simulation settings (π = 0.25 and 0.5) in Fig. S8i,m in order to compare the accuracy of dPC and FDR estimates across a wide range of SNR levels. Similar to our earlier analyses, we also compared dPCA(norep) with dPCA(truevar). The former is the same as dPCA-pnr, the latter corresponds to the dPCA-pnr analysis applied to the same simulation data with no replicate, but after replacing σ^2 by the σ^2 estimated from using the replicate samples. Overall, the simulations yielded similar conclusions as before. dPCA(norep) was relatively more conservative compared to dPCA(truevar). The σ^2 estimates provided by dPCA(norep) tend to be bigger than the σ^2 of dPCA(truevar). For instance, for the π = 0.5 case, the σ^2 estimates and the associated SD from dPCA(truevar) and dPCA(norep) were 0.35 (SD = 0.037) and 0.52 (SD = 0.047) respectively. When π = 0.25, the σ^2 estimates from dPCA(truevar) and dPCA(norep) were 0.36 (SD = 0.034) and 0.42 (SD = 0.032) respectively. As a result, dPCA(norep) produced smaller SNR estimates and more conservative estimates of FDR (Fig. S8i1,15-16,m1,m5,n1,n5). SNR > 5 remained to be a reasonable cutoff for determining which dPCs to report. For example, in Fig. S8i, the SNR for dPC1 produced by dPCA(norep) was close to 10. Here the V_j estimate for dPC1 was accurate, and the FDR estimates were conservative. In Fig. S8m, the SNR for dPC1 was slightly above 5. The corresponding V_j and FDR estimates were reasonable, but began to show biases. In Fig. S8n, the SNR for dPC1 was below 5. The V_j and FDR estimates were inaccurate.

**Functional interpretation of dPCs.** dPCA is primarily a method for objectively detecting and reporting differential patterns and differential loci in the data. After running dPCA, users may want to interpret the biological meanings of the reported dPCs. They may do it using multiple approaches. First, one can directly look at the covariance pattern of multiple proteins in each reported dPC. In the MYC and promoter analyses, for instance, the patterns in V_j readily suggest that dPC1 corresponds to a gene activation module, and dPC2 corresponds to a gene repression module. In the ASB analysis, the pattern in V_j reveals that allele-specific binding of multiple proteins involved in gene activation are correlated and change in the same direction. Second, one may also use other existing ‘omics data to help with the interpretation. For instance, analyses of gene set enrichment in the MYC and promoter data were also able...
to link dPC1 to gene activation, and link dPC2 to Polycomb Repressive Complex 2 and gene repression. Gene set enrichment is only an example to demonstrate how external data may be used to help with interpretation. In principle, many other types of omics data such as gene ontogeny, DNA motif, existing gene expression data in public databases, etc., may also provide useful information for interpreting dPCs.

Third, in real applications, investigators may also examine the findings through functional experiments. In this regard, one can choose some representative differential loci from each dPC and then study them in detail using various types of functional validation experiments.

In practice, a combination of the above methods may be used to interpret dPCs. dPCs can be interpreted and used both separately and jointly. For example, in our MYC and promoter analyses, the observation that dPC2 is associated with H3K27me3 and PRC2, and therefore has a repressive function does not require one to know what dPC1 is. Similarly, without dPC2, one can also link dPC1 to gene activation. On the other hand, when dPC1 and dPC2 were analyzed jointly, we were able to better explain differential gene expression and better characterize how differential chromatin patterns are associated with different gene sets.

In real applications, whether an investigator wants to use dPCs separately or jointly depends on his/her specific research goals and available resources. For instance, if one only wants to roughly know what each pattern is, one could pick up some representative loci from each pattern and do functional studies for each pattern separately. On the other hand, if one wants to explain differential signals in another independent dataset (e.g., differential RNA-seq), one may want to use dPCs jointly to explain as much differences as possible. Using dPCs separately may lose some information, but it makes the analysis simpler and easier. Using dPCs jointly may produce additional information, but may raise the cost of the follow-up experiments since one may need to study a larger number of patterns and loci. This is similar to how biologists study functions of two genes in a pathway. When there is only limited amount of resources, one can first study each gene separately. As more knowledge is accumulated and more resources become available, one can then study their interactions and joint effects using more sophisticated and expensive experiments.

In our MYC and promoter analyses, dPC1 was found to strongly correlate with the benchmark test signals (i.e., differential MYC binding and differential gene expression), whereas dPC2 was found to have much weaker correlation. The differential correlation levels mainly reflect the underlying nature of the ChIP-seq data for gene activation marks and for the gene repression mark H3K27me3. In both examples, dPC1 was mainly driven by gene activation marks including H3K4me1, H3K4me2, H3K4me3, H3K9ac and H3K27ac. dPC2 was mainly driven by the gene repression mark H3K27me3. Figure S1J shows that the differential H3K27me3 ChIP-seq signal (data from the Broad Institute) intrinsically had much weaker correlation with MYC ChIP-seq (Pearson correlation $\rho = -0.026$) and RNA-seq ($\rho = -0.11$) compared to the correlation between these test signals and individual gene activation marks (e.g., compare Fig. S1J with Fig. S1d,e). In fact, in Example I, when we examined the correlation between the differential MYC ChIP-seq and each individual gene activation histone mark that drives dPC1, the smallest correlation was $\rho = 0.41$. Similarly, for differential RNA-seq in Example II, the smallest correlation was $\rho = 0.44$. As a comparison, the two H3K27me3 datasets both had much weaker correlation with the differential MYC binding ($\rho = -0.026$ and $-0.067$ for BR and UW respectively) and differential RNA-seq ($\rho = -0.11$ and $-0.13$ for BR and UW). Thus, the gene activation marks individually all had stronger correlation with the test signals than H3K27me3 did. dPCA integrates information from multiple datasets. By doing so, each dPC will reflect the underlying characteristics of the datasets that drive this dPC. Therefore, it is not surprising that dPC2 had much weaker correlation with the test signals compared to dPC1.

One may further ask why differential H3K27me3 only weakly correlates with differential MYC binding or RNA-seq, but this now becomes a general biology question rather than a question tied to a specific computational method. There could be many possible explanations for this weak correlation. Below are a few examples. First, differential H3K27me3 may correlate with other genomic signals. For instance, the MYC motif is an E-box motif which can be recognized by many other TFs. Therefore it is possible that differential H3K27me3 at certain motif sites marks differential binding of other TFs. This may dilute the correlation between H3K27me3 and MYC binding when the E-box motif sites in the genome are examined together, leading to a weak marginal correlation. Conversely, differential MYC binding may correlate with other chromatin signals. This may dilute the observed MYC-H3K27me3 correlation as well. Second, it is possible that other factors are required to translate differential H3K27me3 to differential MYC binding or RNA-seq. H3K27me3 is a repressive mark. Since it does not directly activate gene expression or enhance activity, both high and low H3K27me3 may correspond to low expression in the absence of gene activation signal. As a result, differential H3K27me3 does not necessarily mean differential expression. Third, it is also possible that differential H3K27me3 intrinsically only has a small effect (thus weak correlation) in terms of changing MYC binding or RNA-seq at each locus. However, since it can affect many genomic loci, the cumulative effects of many small changes of MYC binding or RNA-seq may lead to a biologically significant outcome.

In general, it is not easy to identify the exact cause for a weak correlation. To do so, one has to examine a number of different explanations and rule out many possibilities. This is not only non-trivial but may also be unrealistic since one does not always have information on all factors that affect a system. Therefore, while there are many possible explanations for the observed weak correlation between H3K27me3 and differential MYC binding or RNA-seq, identifying the exact reason is beyond the scope of the current paper.

One might think that the difficulty to find the exact reason to interpret the weak correlation between dPC2 and the test signals in our MYC and promoter analyses represents a limitation of dPCA, that is, it is difficult to interpret the meta-ChIP patterns represented by dPCs. However, we would like to point out that, in our examples, the difficulty in explaining the weak correlations was not caused by the dPCA method. Instead, it was determined by the data. For example, even without using dPCA, one can analyze each individual dataset separately. One will still observe the weak correlation between the differential H3K27me3 and the differential MYC binding or RNA-seq without knowing why they should be weakly correlated. Thus, the difficulty in the interpretation is intrinsic in the data rather than caused by combining multiple datasets into a meta-ChIP pattern. If it represents an interpretation problem, it is an interpretation problem tied to all methods analyzing these data, rather than a problem specific to dPCA.

This is different from a situation where analyzing each individual dataset separately can give results with clear biological interpretation, but analyzing them jointly using dPCA produces results that are not interpretable. Clearly, this is not the case here. In our analyses, dPCA did what it is supposed to do.
to do, that is, to objectively report differences in the data after synthesizing correlated data to obtain a more concise description of the differences. The intrinsic nature of the data is retained in the reported meta-ChIP patterns. In fact, our results for dPC1 (i.e., the gene activation module) show that, instead of causing troubles, using the meta-ChIP pattern was able to improve the characterization of biological differences compared to analyzing each dataset separately.

dPCA detects differences in the data it analyzes. It does not directly detect differences in the data that it does not analyze, such as the benchmark MYC ChIP-seq and RNA-seq data in our examples, which may or may not correlate with the ChIP-seq data analyzed by dPCA. In reality, meaningful benchmark data may not always be available, and we may not know everything about a biological system. As a result, while a strong correlation between an independent benchmark dataset with the dPCA results (e.g., dPC1) shows that dPCA is capable of detecting meaningful differences, a weak correlation between a test dataset and dPCA results (e.g., dPC2) does not mean that the differences detected by dPCA are unreal, since the weak correlation could reflect the lack of relevant benchmark data or our limited knowledge about the biology. In fact, based on our simulations that took into account the real data characteristics, the dPC2 reported in the MYC and promoter analyses is very likely to represent a true differential pattern given its signal-to-noise ratio, and the differential loci reported along dPC2 are very likely to represent true differences not explained by the cross-replicate variability. In future applications, differences without strong correlation with other existing data may represent things people do not understand well, and may lead to new discoveries. Investigators therefore may want to study them further (e.g., through functional experiments on representative loci for each dPC).

In our examples, dPC1 had a stronger correlation with MYC binding and RNA-seq. In general, however, dPC1 may not necessarily have stronger correlation with an independent dataset compared to the other dPCs. Different dPCs represent different types of differential signals (e.g., activation vs. repression) and therefore may correlate with different things. Depending on what test signal is used for studying correlation, it is possible that other dPCs correlate with a test signal better than dPC1 does. On the other hand, dPCs characterize synergistic patterns of PDI in multiple datasets. Within a single dPC, correlated changes of individual datasets are integrated based on the synergy pattern the dPC represents. This often strengthens the signal carried by each individual dataset. As a result, if this synergy pattern correlates with a test signal related to the biological function the synergy represents, the dPC often will correlate with the test signal better (or not worse) than each individual dataset that drives this dPC, as shown by our ranking analyses in the examples.

Objectively detecting differences in ChIP-seq data and functional interpretation of the detected differences are two related but different problems. They require development of different methods. For example, for detecting differences, one needs methods such as t-test or dPCA. In contrast, for functional interpretation, one needs methods such as gene set enrichment analysis or gene ontology analysis, etc. dPCA belongs to the former category and is primarily designed as a method for objectively reporting differences in the data rather than directly interpreting functions of differences. Correspondingly, this article primarily focused on testing the method’s properties related to the detection of differences. While we discussed several ways to interpret functions of dPCs, functional interpretation in general still represents an open problem. As we discussed before, many questions in functional interpretation, such as how to interpret a weak correlation between the detected differences with another independent dataset, are not specific to dPCA but general for all methods that analyze differential binding. Therefore it is important to continually develop new methods for functional interpretation of differential ChIP-seq signals.

We conclude by pointing out that knowing the biological functions of each dPC is not a necessary condition for dPCA to be useful. The correlation structure among multiple proteins revealed by the exploratory dPCA analysis itself may represent new knowledge and can be useful for various purposes. For example, even without knowing the function of the dPC1 in the ASB analysis, the multi-protein correlation pattern revealed by this dPC already provides new knowledge that allele-specific binding of multiple proteins is correlated and these proteins tend to be skewed toward the same allele. Subsequently, computational biologists can use this knowledge to develop better algorithms and tools that take advantage of this correlation to improve the detection of allele-specificity.

Fig. S1. Data structure and exploratory plots. (A)-(B): Limitations of using binary peak calls to study differential binding. (A) The ENCODE H3K4me3 ChIP-seq data from the Broad Institute (BR) were analyzed using CisGenome. The venn diagram approach identified three peak categories: 1848 Huvec-specific, 1447 K562-specific and 10135 common peaks. Peaks were grouped into six bins based on the log2 fold change ($\log_2 FC$) of quantitative ChIP-seq signals between K562 and Huvec. For each peak category, the frequency of peaks in different fold change bins is shown (e.g., about 60% of Huvec-specific peaks had $\log_2 FC < -2$). The plot shows that the signs of $\log_2 FC$ are not always consistent with the condition-specificity classification. (B) For the H3K4me3 data in (A), this plot shows the percentage of peaks explained by each peak category within each fold change bin. About 20% of the peaks with $|\log_2 FC| > 2$ were called as “common peaks” by the venn diagram approach.

(C) Illustration of the dPCA data structure. dPCA considers two conditions (e.g., two cell types). Each condition has multiple datasets. Each dataset has one or more replicates. A list of candidate loci is given (blocks shown at the bottom). dPCA will analyze quantitative differences between the two conditions at these loci. (D)-(E): Illustration of the type of data considered by dPCA. dPCA considers datasets in which protein-DNA interaction changes ($d_{gm}$) between two conditions are bidirectional and fluctuate around zero. (D) A scatter plot showing MYC ChIP-seq $d_{gm}$ vs. H3K27ac ChIP-seq $d_{gm}$ at the 58,997 MYC motif sites analyzed in Example I in the main article. Each dot is a motif site. The mean of $d_{gm}$s in each dataset, $\bar{d}_{gm}$, is shown both on top of the plot and as a purple star in the plot. Both $\bar{d}_{gm}$s are only slightly different from zero. ‘Cor’: Pearson correlation coefficient. (E) A scatter plot showing RNA-seq $d_{gm}$ vs. H3K4me1 ChIP-seq $d_{gm}$ at the 22,368 human promoters analyzed in Example II in the main article. Again, both $\bar{d}_{gm}$s are only slightly different from zero. (F)-(G): Plots similar to (D) and (E), but are obtained after subtracting control intensities $d_{g,control}$ from ChIP intensities $d_{gm}$ to obtain $\hat{d}_{gm} = d_{gm} - d_{g,control}$. (F) MYC ChIP-seq $\hat{d}_{gm}$ vs. H3K27ac ChIP-seq $\hat{d}_{gm}$ at the 58,997 motif sites in the MYC analysis. (G) RNA-seq $\hat{d}_{gm}$ vs. H3K4me1 ChIP-seq $\hat{d}_{gm}$ at the 22,368 promoters in the promoter analysis. (F) and (G) illustrate that subtracting control intensities deteriorates the quantitative nature of the signals, as indicated by the decreased correlation coefficient between related datasets (i.e., H3K27ac and MYC ChIP-seq, H3K4me1 ChIP-seq and RNA-seq) compared to (D) and (E). This is one reason why we treat controls separately from ChIP datasets (see Text S1). (H)-(I): Box plots comparing the variability of $d_{gm}$ (mean centered to zero) in different datasets in the MYC (H) and Promoter (I) analyses. These plots show that the Input control datasets have smaller variance than most TF/HM ChIP datasets, which is expected since control data do not contain true biological variation of interest. This explains why dPCA, by default, does not scale columns of $\mathbf{D}$ to have equal unitary variance. Since $d_{gm}$s are at log2 scale, scaling column variances may artificially increase the fraction of data variation contributed by datasets in which there is little or no real difference between the two conditions.

(J) Scatter plots showing MYC ChIP-seq $d_{gm}$ vs. H3K27me3 (BR) ChIP-seq $d_{gm}$ at the 58,997 MYC motif sites in Example I, and RNA-seq $d_{gm}$ vs. H3K27me3 (BR) ChIP-seq $d_{gm}$ at the 22,368 promoters in Example II. These plots show that H3K27me3 has weak negative correlation with MYC binding and RNA-seq.


Fig. S2. Analyses of motif sites in Example I. (A)-(K). dPCA results for the MYC analysis. (A) The cumulative number of differential loci (5% FDR) reported by dPCA (dPCA-P) as the number of dPCs increases (B)-(C). The numbers of differential loci and gold standard (GS) MYC differential binding sites detected by dPC1 and dPC2 at the 5% FDR cutoff from five variants of dPCA (A, P, PR0.5, PR0.75, Z) that handle absolute binding differently are shown in (C). The venn diagrams in (B) further display the numbers from two representative methods, P (numbers not in the brackets) and PR0.5 (numbers in the square brackets). P (i.e., dPCA-P) is the method used in the main article. For each method, the numbers of loci reported by each dPC alone and by both dPCs are shown. For PR0.5 and PR0.75, ‘dPC1&dPC2’ are loci with FDR ≤ 0.05 and R2dPC1 > c (c = 0.5 or 0.75) for both dPCs. ‘dPC1-dPC1&dPC2’ are loci with (FDR ≤ 0.05 and R2dPC1 < c) for dPC1 but not satisfy (FDR ≤ 0.05 and R2dPC1 > c) for dPC2. These may contain loci with (FDR ≤ 0.05 and R2dPC2 < c) for dPC2 and dPC1 & dPC2 not differential; ‘+/-’ dPC1 up in K562 and dPC2 down in K562; etc.). For each class, the distribution and the mean of the MYC ChIP-seq log2 fold change (|log2 FC|) for each class is shown in (E). Similar to (E), one can use different methods (A, PR0.5, PR0.75, Z) to define the 9 classes. For each class defined by each method, the mean MYC ChIP-seq |log2 FC| is shown in (F) as a heat map. Both (E) and (F) show that on average, the classes in which dPC1 and dPC2 had opposite signs (i.e., the classes labeled as ‘+/-’ or ‘-/+’) had the largest mean |log2 FC| compared to the other classes.

The first two PCs discovered by applying PCA to dPCA-P using the 58997 MYC motif sites. The numbers in the square brackets are the percentages of variance explained.

Top enriched gene sets identified for four dPC classes: dPC1+, dPC1-, dPC2+ and dPC2-. dPC classes are defined using dPCA-P and PR0.5 respectively. For each method, the heat map on the left shows the log2 fold enrichment for the corresponding dPC class (i.e., Ksc) and the number of genes found in the dPC class (i.e., ksc) are displayed in the brackets as ‘(ksc/Ksc)’. The bar shows the log2 fold enrichment (rsc, see Text S1 for definition) of gene set s in dPC class c compared to random expectation. Statistical significance is indicated by *s, **s and ***s represent q-value<0.1, 0.01 and 0.001 respectively. (I) For each gene set in (H), the heat map shows its log2 fold enrichment in the corresponding dPC class defined by several other methods (A, PR0.5, PR0.75, Z) that use absolute binding differently. Red means enriched (i.e., rsc > 0). Blue means depleted (i.e., rsc < 0). (J) This heat map shows whether the enrichments in (I) are statistically significant (q-value<0.1) or not. Red means significant. Blue means not significant. (K) Top enriched gene sets identified for 9 (dPC1,dPC2) classes. dPC classes are defined using dPCA-P and PR0.5 respectively. For each method, the heat map on the left shows the log2 fold enrichment (i.e., rsc) for each gene set s in each class c. Red means enriched, and blue means depleted. The heat map on the right shows whether the enrichment of gene set s in class c was statistically significant (i.e., q-value<0.1). Red means significant. The number of genes of each gene set is shown in the brackets. For each gene set, the two smallest q-values across 9 classes defined by P and PR0.5 respectively are shown in the square brackets ([P/PR0.5]). Some dPC classes did not have enriched gene sets. (L) (N): dPCA analysis of SRF, GABP and E2F6 motif sites (Text S1). (L) SRF in Hepg2 and K562. (M) GABP in Hepg2 and K562. (N) E2F6 in K562 and Helas3. For each TF, the figure has four panels. (1) The first two dPCs. (2) Estimated SNR by each dPC. (3) Differential TF binding measured by log2 ChIP-seq fold change for the TF in question was plotted against dPC1 (β1) and dPC2 (β2) respectively. ‘Cor’: Pearson correlation coefficients. (4) The numbers of top ranked motif sites that are truly differentially bound by the TF are shown for different ranking methods. Legend for the methods is the same as the legend in (D).
Fig. S3. Simulation results based on real data characteristics. Simulation data were generated using residuals from each of the three real data examples (MYC, Promoter, ASB). For each case, simulations were performed under different global signal-to-noise ratio (SNR) settings with the overall SNR level controlled by a parameter $\pi_0$ (see Text S1). Increasing $\pi_0$ will increase the SNR. (A)-(B): results for MYC. (C)-(E): results for promoter. (F)-(H): results for ASB. Each plot has 7 columns. From left to right, they are (1) estimated signal-to-noise ratio for each dPC; (2) accuracy of $v_j$ estimates, measured by the cosine distance; (3) $\hat{\lambda}_j - \lambda_j$, error of eigenvalue estimates; (4) cumulative percentage of variance explained by top dPCs (dPCA) or PCs (PCA); (5)-(7) the true FDR at different levels of the estimated FDR for the first three dPCs. All plots show the average performance of 10 simulations. Vertical bars indicate ±1 standard deviation of the 10 simulations.
Variability and Reproducibility of dPCs

(A) MYC  
B Promoter  
C ASB

Locus- and Dataset-specific Variances

D

E

F

G

H

I

J

K

L

Variability and Reproducibility of dPCs

(A)–(C): Examples showing that dPC estimates cannot be reproduced when SNR is small. For each of the three examples – (A) MYC, (B) Promoter and (C) ASB – a pair of simulations was performed to mimic the situation in which two labs independently generate similar data and then analyze the data. Simulation data were created using the same procedure used to generate Figs. 3 and S3 (under the $\pi_0 = 0.25$ setting). Data within each pair of simulations were generated from the same underlying true model. The plots show that the distance between the $\hat{v}_j$s from the two simulations increases as SNR decreases (see SNRs in Figs. 3, S3). This suggests that dPCs with small SNR cannot be reproduced between two independent labs.

(D): Comparison between dPCA based on the equal variance assumption and variants of dPCA based on using (1) dataset-specific, (2) locus-specific, and (3) locus- and dataset-specific variances. For (2) and (3), the methods were tested both with and without variance shrinking. For each of the three real data examples – MYC, Promoter and ASB – we evaluated the performance of each method for ranking differential loci, similar to Figs. 2e, 4d, 5(c,d). Using the locus- and/or dataset-specific variances did not clearly improve the original dPCA based on the equal variance.

(E)-(I): Assessment of using t-distribution as the null. (E) shows QQ-plots for checking normality of residuals. Empirical quantiles of residuals $\epsilon_{gimk} = x_{gimk} - \bar{x}_{gim}$ from each dataset $m$ and condition $i$ were plotted against theoretical quantiles from a normal distribution. A few examples from MYC, Promoter and ASB analyses are shown. In the ASB plots, SNPs with no mapped reads were excluded from the analyses. (F) shows residual $\epsilon_{gimk}$ vs. $d_{gm} = \bar{x}_{g1m} - \bar{x}_{g2m}$ in a dataset $m$. (G) shows $s_g = \sqrt{\sum_{r=1}^{K_m}(x_{gimk} - \bar{x}_{gim})^2/\sum_{r=1}^{K_m}(K_m - 1)}$ vs. $\beta_{gj}$ for each reported dPC in MYC and Promoter data. For ASB, $s_g = \sqrt{\sum_{r=1}^{K_m}(y_{gimk} - \bar{y}_{gim})^2/\sum_{r=1}^{K_m}(K_m - 1)}$ vs. $\beta_{gj}$ is shown. The red line shows the moving average of $s_g$ in a sliding window moving across $\beta_{gj}$. The window size corresponds to 10% of the total number of loci. (H) Loci in (G) are grouped into bins based on $\beta_{gj}$. For each bin, seven summary statistics are computed: mean (mu), standard deviation (sd), and the 5th, 25th, 50th, 75th, and 95th percentiles of $s_g$. For each summary statistic, (H) shows the coefficient of variation (CV) across the bins. The small CVs show that each summary statistic can be roughly viewed as a constant across different $\beta_{gj}$ values. (G) and (H) show that the numerator and the denominator in the t-statistic $T_{gj}$ are roughly independent. (I) shows the histogram of p-values for each dPC computed using the t-distribution as the null. The right tail of each plot is approximately uniform, without alerting signs for serious problems in model assumptions.

(J)-(L): dPCs discovered by two variants of dPCA, dPCA-A and dPCA-Z. dPCA-A applies dPCA to all loci (e.g., all 138,325 MYC motif sites) regardless of their binding status. dPCA-Z sets differences in non-peak loci to zero before dPCA. (J) shows the dPCs discovered in the MYC, Promoter and ASB analyses by dPCA-A. (K) shows the dPCs discovered by dPCA-Z. (L) shows the signal-to-noise ratio for each dPC in these three examples.
negligible effects on the dPCA results. It also illustrates that centering column means to zero has negligible effects on dPCA in applications where the setting of dPCA which centers column means to zero. These two analyses produced similar results. In our data, RNA-seq fold change versus dPC1 (\(N\)) The first two dPCs (BR: Broad; UW: Univ. of Washington; DK: Duke; NC: Univ. of North Carolina). (O) Estimated dPCA (dPCA-P) analysis without centering columns across 9 classes defined by P and PR0.5 respectively are shown in the square brackets ([P/PR0.5]). Some dPC classes had fewer than three enriched gene sets. (K) Statistical significance of each gene set (i.e., \(q\)-value \(\leq 0.1\)). Red means significant. The number of genes of each gene set is shown in the brackets. For each gene set, the two smallest \(q\)-values were grouped into 9 classes based on dPC1 and dPC2 differential status ('+': up in K562; '-': down in K562; 'o': non-differential). For each class, the distribution and the mean of the RNA-seq log2 fold change (\(\log_{2} FC\)) are shown in (G). Similar to (G), one can use different methods (A, PR0.5, PR0.75, Z) to define the 9 classes. For each class defined by each method, the mean RNA-seq \(\log_{2} FC\) is shown in (H) as a heat map. Both (G) and (H) show that on average, the classes in which dPC1 and dPC2 had opposite signs (i.e., the classes labeled as '+/-' or '-/+') had the largest magnitude of mean \(\log_{2} FC\) compared to the other classes. (I) For each method in (H), this heat map shows the Pearson's correlation coefficient between dPC1 and DE. Again, the correlation is the strongest when dPC1 and dPC2 had opposite signs. Statistical significance is indicated by stars. '*', '**' and '***' represent \(q\)-value \(\leq 0.05\) and \(\leq 0.01\) and \(\leq 0.001\) respectively. (J) Top enriched gene sets identified for four dPC classes: dPC1+, dPC1-, dPC2+ and dPC2-. dPC classes are defined using dPCA-P. 'Cor': Pearson correlation coefficients. (Q) The numbers of top ranked promoters that are true DE are shown for different ranking methods. 'dPC1' and 'dPC2': rankings along dPC1 and dPC2 by dPCA-P. 'Single PU': single dataset ranking of the 22368 promoters analyzed by dPCA. 'Single A': single dataset ranking using all 24376 promoters. 'Single P': single dataset ranking using promoters bound in each dataset (see details in Text S1). (E)-(F): The numbers of differential promoters and gold standard (GS) promoters (i.e., those with DE) detected by dPC1 and dPC2 at the 5% FDR cutoff from five variants of dPCA (A, P, PR0.5, PR0.75, Z) that handle absolute binding differently are shown in (F). The vein diagrams in (E) further display the numbers from two representative methods, P (numbers not in the brackets) and PR0.5 (numbers in the square brackets). P (i.e., dPCA-P) is the method used in the main article. For each method, the numbers of loci reported by each dPC alone and by both dPCs are shown. For PR0.5 and PR0.75, 'dPC1&dPC2' are loci with FDR \(\leq 0.05\) and \(R^{D} > c\) (i.e., \(c = 0.5\) or 0.75) for both dPCs. 'dPC1-dPC1&dPC2' are loci with (FDR \(\leq 0.05\) and \(R^{D} \leq c\)) for dPC1 but did not satisfy (FDR \(\leq 0.05\) and \(R^{D} > c\)) for dPC2. These may contain loci with (FDR \(\leq 0.05\) and \(R^{D} \leq c\)) for dPC2. 'dPC2-dPC1&dPC2' is defined similarly. (G)-(H): Using dPCA-P, promoters were grouped into 9 classes based on dPC1 and dPC2 differential status ('+': up in K562; '-': down in K562; 'o': non-differential). For each class, the distribution and the mean of the RNA-seq log2 fold change (\(\log_{2} FC\)) are shown in (G). Similar to (G), one can use different methods (A, PR0.5, PR0.75, Z) to define the 9 classes. For each class defined by each method, the mean RNA-seq \(\log_{2} FC\) is shown in (H) as a heat map. Both (G) and (H) show that on average, the classes in which dPC1 and dPC2 had opposite signs (i.e., the classes labeled as '+/-' or '-/+') had the largest magnitude of mean \(\log_{2} FC\) compared to the other classes. (I) For each method in (H), this heat map shows the Pearson's correlation coefficient between dPC1 and DE. Again, the correlation is the strongest when dPC1 and dPC2 had opposite signs. Statistical significance is indicated by stars. '*', '**' and '***' represent \(q\)-value \(\leq 0.05\) and \(\leq 0.01\) and \(\leq 0.001\) respectively. (K) Top enriched gene sets identified for four dPC classes: dPC1+, dPC1-, dPC2+ and dPC2-. dPC classes are defined using dPCA-P. (Text S1) For each gene set, the total number of genes in the set (i.e., \(K_{s}\)) and the number of genes found in the dPC class (i.e., \(K_{s,c}\)) are displayed in the brackets as \([K_{s,c}/K_{s}]\). The bar shows the log2 fold enrichment (\(r_{sc}\), see Text S1 for definition) of each gene set \(s\) in dPC class \(c\) compared to random expectation. Statistical significance is indicated by stars. '*' and '***' represent \(q\)-value \(\leq 0.01\) and 0.001 respectively. (K) For each gene set in (J), the heat map shows its log2 fold enrichment in the corresponding dPC class defined by several other methods (A, PR0.5, PR0.75, Z) that use absolute binding differently. Red means enriched (i.e., \(r_{sc} > 0\)). Blue means depleted (i.e., \(r_{sc} < 0\)). (L) This heat map shows whether the enrichments in (K) were statistically significant (\(q\)-value \(\leq 0.1\)) or not. Red means significant. Blue means not significant. (N) Top enriched gene sets identified for 9 (dPC1&dPC2) classes. For each method, the map on the left shows the log2 fold enrichment (i.e., \(r_{sc}\)) of each gene set \(s\) in each class \(c\). Red means enriched, and blue means depleted. The heat map on the right shows whether the enrichment of gene set \(s\) in class \(c\) was statistically significant (i.e., \(q\)-value \(\leq 0.1\)). Red means significant. The number of genes of each gene set is shown in the brackets. For each gene set, the two smallest \(q\)-values across 9 classes defined by P and PR0.5 respectively are shown in the square brackets ([P/PR0.5]). Some dPC classes had fewer than three enriched gene sets. (N)-(Q): dPCA (dPCA-P) analysis without centering column means of D to have zero mean. As a comparison, Fig. 4 in the main article shows the results based on the default parameter setting of dPCA which centers column means to zero. These two analyses produced similar results. In our data, \(d_{\text{null}}\) within each dataset fluctuate around zero, however the mean of \(d_{\text{null}}\) in each dataset is not exactly zero. The fact that (N)-(Q) and Fig. 4 are similar demonstrates that the slight deviation of \(d_{\text{null}}\) from zero has negligible effects on the dPCA results. It also illustrates that centering column means to zero has negligible effects on dPCA in applications where the mean of the RNA-seq fold change versus dPC1 (\(\beta_{12}\)) and dPC2 (\(\beta_{22}\)). 'Cor': Pearson correlation coefficients. (Q) The numbers of top ranked promoters that are true DE are shown for different ranking methods.
**Fig. S6.** dPCA analysis of ASB in Example III. (A) The first two dPCs, explaining 47.3% and 14.9% of variance respectively. However, only dPC1 passes the $SNR > 5$ cutoff and will be reported. (B) Histograms showing the distributions of $\hat{T}_{gj}$ for dPC1 and dPC2 respectively. (C) Box plots showing the $\hat{T}_{gj}$ distributions for each dPC. (D) The cumulative number of differential loci reported by dPCA (dPCA-P) as the number of dPCs increases. (E) The correlation between ChIP-seq ASB ($\hat{\beta}_{g1}$) and RNA-seq ASE ($\log_2 FC$) for chrX (non-pseudautosomal chromosome X) SNPs that can be annotated with heterozygote exonic SNPs. Negative values mean that the reference allele is expected to be active in the ChIP-seq data. SNPs can be grouped into four classes based on the signs of their ASB and ASE data. The number of SNPs in each class is also shown. 'Cor.': the Pearson correlation between ASB and ASE. The plot demonstrates that there is no obvious reference bias, and ASB and ASE are positively correlated. (F) The numbers of top ranked SNPs that are chrX SNPs are shown for different ranking methods. dPCA is compared with three single dataset based methods including 'Single A' (all loci), 'Single P' (peak loci) and 'Single PU' (union of peak loci). (G) The numbers of top ranked SNPs that are in the neighborhood of exonic ASE SNPs. (H) The numbers of top ranked autosomal SNPs that are in the neighborhood of autosomal exonic ASE SNPs. (I)-(K): Similar to (F)-(H), but now dPCA was applied to the 9 datasets from the Broad Institute. (L) Top enriched gene sets identified for dPC1. Here the dPC class is defined using dPCA-P. For each gene set, the total number of genes in the set (i.e., $K_s$) and the number of genes found in the dPC1 class (i.e., $k_{sc}$) are displayed in the brackets as "($k_{sc}/K_s$)". The bar shows the log2 fold enrichment ($r_{sc}$, see Text S1 for definition) of gene set $s$ in dPC class $c$ (here dPC1) compared to random expectation. Statistical significance is indicated by stars. '*', '**' and '***' represent $q$-value $\leq 0.1$, 0.01 and 0.001 respectively. (M) For each gene set in (L), the heat map shows its log2 fold enrichment in dPC1 defined by several other methods (A, PR0.5, PR0.75, Z) that use absolute binding differently. Red means enriched (i.e., $r_{sc} > 0$). Blue means depleted (i.e., $r_{sc} < 0$). (N) This heat map shows whether the enrichments in (M) were statistically significant ($q$-value $\leq 0.1$) or not. Red means significant. Blue means not significant. Here all enrichments were significant.
Fig. S7. Absolute binding. (A) Examples to illustrate $R^2$. A few differential loci along dPC1 and dPC2 in the MYC analysis are shown with their ranks and $R^2$ values. For each locus, the left plot shows the binding intensities in K562 and Huvec in each dataset. ‘*’ on top of a dataset means there is a binding peak call in that dataset in at least one cell type. The right plot shows the MYC ChIP-seq binding intensity. Gold ‘*’ indicates that the locus has differential MYC ChIP-seq $|\log_2 FC| > 1.5$. (B) For each dPC, differential loci (5% FDR) are grouped into four bins: $R^2 \in [0.0,0.25],[0.25,0.5],[0.5,0.75],[0.75,1]$. For each bin, the percentage of loci that are associated with binding peak calls in each dataset (in at least one cell type) is shown. (C)-(D): For each dPC, differential loci are stratified based on their ranks in dPCA. Each stratum contains 1000 loci. (C) and (D) are two representative strata. Loci in (C) have rank $\in (9000,10000]$. Loci in (D) have rank $\in [25000,26000]$. Within each stratum, differential loci are grouped into four $R^2$ bins as above. For each bin, the mean of $\bar{a}_{gm} = \frac{\bar{x}_{g1m} + \bar{x}_{g2m}}{2}$ is computed for each dataset $m$ using all differential loci in that bin. The mean $\bar{a}_{gm}$ is shown for each dataset and each bin. (E) For each example (MYC, Promoter, ASB) and each dPC, the plot shows the percentages of the top $N$ loci that fall into the four $R^2$ bins. The percentages are shown as a function of $N$. (F)-(G): For each data example, different methods for handling the absolute binding information are compared. (F) shows $X_N$, the number of gold standard loci detected by the top $N$ differential loci, as a function of $N$. The plots only show differential loci that passed the 5% FDR cutoff in the MYC, Promoter and ASB-chrX analyses, or 25% FDR in the ASB-ASE analysis (due to small number of SNPs associated with ASE). (G) shows $X_N$ versus $1 - X_N/N$ (i.e., 1-Positive Predictive Value (PPV)). (H)-(I): An example to illustrate why increasing $R^2$ may not necessarily improve ranking performance compared to dPCA-P. Differential loci for dPC1 in the MYC analysis are ranked by dPCA-P. For each of the four methods – P, PR0.25, PR0.5 and PR0.75 – (H) shows the PPV of the top ranked loci with $-|T_{gj}| < t'$. The PPV is plotted as a function of $t'$. Increasing $R^2$ increases PPV but decreases the sensitivity, and therefore may not necessarily improve the overall receiver operating characteristics. (J) For each data example in (F), the numbers of differential loci and gold standard (GS) loci detected by different methods at 5% FDR (or 25% FDR for ASB-ASE) are shown. Each dPC has three bars. The top bar shows how the loci number increases when one moves from PR0.75 to PR0.5, PR0.25 and P. The middle bar shows how the loci number increases from APR0.75 to APR0.5, APR0.25, AP and A. The bottom bar shows how the loci number reported by Z. For example, in the left most bar plot, the top bar shows the numbers of differential loci along dPC1 reported by PR0.75, PR0.5, PR0.25 and P in the MYC analysis. For the ASB data, the GS are based on RNA-seq ASE. The results based on chrX GS are similar and therefore not shown here due to the space constraint. (K) Comparison of the consistency between different pairs of methods. For each pair of methods X and Y (denoted by X-Y), each data example and dPC, the figure shows two bars. The top bar is the percentage of loci reported by X that are also discovered by Y, both at the 5% FDR level. The bottom bar is the percentage of loci reported by Y that are also discovered by X. Here 1 means 100%. For example, the overlap between APR0.75 and PR0.75 is almost 100% in the Promoter data.
Fig. S8. The modified dPCA applied to data without replicate samples. (A)-(C): dPCA-nr applied to the MYC data. (A) shows the top two dPCs. (B) shows the signal-to-noise ratio. (C) shows the ranking performance of dPCA-nr compared to individual datasets. (D)-(F): similar to (A)-(C), but for the promoter analysis. (G)-(I): similar to (A)-(C), but for the ASB analysis using dPCA-pnr. (J)-(K): Simulation studies for testing dPCA-nr based on the real data characteristics in the promoter example. Simulations were done in different global SNR levels controlled by the parameter $\pi_0$. Two representative simulation settings are shown. In each plot, the black curve (i.e., $dPCA(norep)$) corresponds to dPCA-nr. The blue curve (i.e., $dPCA(truevar)$) corresponds to applying the dPCA-nr to the reduced data without replicate, but substituting the dPCA-nr variance estimate $\hat{\sigma}^2$ by the $\hat{\sigma}^2$ estimate obtained from using replicate samples. Each plot has seven columns. From left to right, they are (1) estimated signal-to-noise ratio for each dPC; (2) accuracy of $v_j$ estimates, measured by the cosine distance; (3) $\hat{\lambda}_j - \lambda_j$, error of eigenvalue estimates; (4) cumulative percentage of variance explained by top dPCs (dPCA) or PCs (PCA); (5)-(7) The true FDR at different levels of the estimated FDR for the first three dPCs. All plots show the average performance of 10 simulations. Vertical bars indicate $\pm 1$ standard deviation of the 10 simulations. (L)-(N): Simulation studies for testing dPCA-pnr based on the real data characteristics in the ASB example. Three representative simulation settings are shown. The meanings of plots are similar to those in (J) and (K).
Table S1. Summary of datasets

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Lab†</th>
<th>MYC, Promoter‡</th>
<th>SRF, GABP‡</th>
<th>E2F6§</th>
<th>ASB¶</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>K562</td>
<td>Huvec</td>
<td>Hepg2</td>
<td>K562</td>
</tr>
<tr>
<td>CTCF</td>
<td>BR</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>H3K4me1</td>
<td>BR</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>H3K4me2</td>
<td>BR</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>H3K4me3</td>
<td>BR</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>H3K9ac</td>
<td>BR</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>H3K9me1</td>
<td>BR</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H3K27ac</td>
<td>BR</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>H3K27me3</td>
<td>BR</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H3K36me3</td>
<td>BR</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Input</td>
<td>BR</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTCF</td>
<td>UW</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>H3K4me3</td>
<td>UW</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>H3K27me3</td>
<td>UW</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>H3K36me3</td>
<td>UW</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Input</td>
<td>UW</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DNase</td>
<td>DK</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FAIRE</td>
<td>NC</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTCF</td>
<td>UTA</td>
<td></td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Pol2</td>
<td>UTA</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Input</td>
<td>UTA</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>cMyc</td>
<td>UTA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pol2</td>
<td>SYDH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pol2Musig</td>
<td>SYDH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cMyc</td>
<td>SYDH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cFos</td>
<td>SYDH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cJun</td>
<td>SYDH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>SYDH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nfkb</td>
<td>SYDH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input</td>
<td>SYDH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total sample no. for dPCA 35 35 30 30 19 20 44

Gold standard: cMyc UTA 3 1
Input: cMyc UTA 1 1
Gold standard: RNA-seq CalTech 2 2
Gold standard: SRF HAIB 1 2
Gold standard: GABP HAIB 2 2
Input: SRF, GABP HAIB 2 2
Gold standard: E2F6 SYDH 2 2
Input: E2F6 SYDH 2 1
Gold standard: RNA-seq SYDH 1

†BR: Broad Institute; UW: Univ. of Washington; DK: Duke Univ.; NC: Univ. of North Carolina, Chapel Hill; UTA: Univ. of Texas, Austin; SYDH: Stanford/Yale/Davis/Harvard Univ.; CalTech: California Institute of Technology; HAIB: HudsonAlpha Institute.
‡Data used in the analyses of MYC motif sites and human promoters
§Data used in the analyses of SRF and GABP motif sites
¶Data used in the analysis of E2F6 motif sites
*Data used in the analysis of ASB