

Supporting Information for 'Efficient Detection and
Classification of Epigenomic Changes Under Multiple
Conditions' by Pedro L. Baldoni*, Naim U. Rashid*,
and Joseph G. Ibrahim*

*Department of Biostatistics, University of North Carolina at Chapel Hill,
Chapel Hill, North Carolina, USA

Supporting Information A

In this appendix, we present a summary of data accession codes, data pre-processing, and parameter specifications from benchmarked methods.

A1 Data accession codes

Web Table 1: GEO sample accession codes of the analyzed data from the ENCODE Consortium

Cell Line	Control	H3K27me3	H3K36me3	EZH2	H3K4me3	H3K27ac	CTCF	RNA-seq
H1hesc	GSM733770	GSM733748	GSM733725	GSM1003524	GSM733657	GSM733718	GSM733672	GSM758566
HeLaS3	GSM733659	GSM733696	GSM733711	GSM1003520	GSM733682	GSM733684	GSM733785	GSM765402
HepG2	GSM733732	GSM733754	GSM733685	GSM1003487	GSM733737	GSM733743	GSM733645	GSM758575
Huvec	GSM733715	GSM733688	GSM733757	GSM1003518	GSM733673	GSM733691	GSM733716	GSM758563

A2 Data Processing

First, we removed PCR duplicates from the BAM files using SAMTools [5] and converted the resulting indexed and sorted files to BED format using BEDTools [10], as RSEG only accepts such a format as input. Then, the fragment length of each ChIP-seq experiment was estimated using csaw and its functions *correlateReads* and *maximizeCcf*. Finally, using the estimated fragment length, read counts from all cell lines were tabulated for their ChIP replicates using fixed-step and non overlapping windows of size 250bp, 500bp, 750bp, and 1000bp through the R package *bamsignals* [8]. For all methods using window-based approaches (csaw, ChIPComp, diffReps, RSEG, THOR, and epigraHMM), we assessed their performance with different window sizes. See Section D2 for a discussion about results with different window sizes.

All the methods considered in the data applications and simulation study outputted a set of differential genomic regions/windows that were used for benchmark purposes. THOR output a list of differential peaks in BED6+4 format (*narrowPeak*) with adjusted p-values. RSEG output a WIG file with genomic windows and their posterior probabilities for differential enrichment. diffReps output an annotated TXT file with differential regions of enrichment and their adjusted p-values. DiffBind output a TXT file with differential regions of enrichment and their respective multiple testing corrected FDR. diffReps output a TXT file with differential regions of enrichment and their p-values. csaw output a TSV file with differential regions of enrichment and their FDR adjusted p-values. For a fair FDR thresholding comparison, we control the total FDR and output the differential regions of enrichment based on the set of posterior probabilities as described in the main text. For a comparison between the Viterbi and the FDR thresholding approach, see Section B3.

The following parametrization was used when calling peaks from the benchmarked methods:

- THOR: *rgt-THOR 'config' -name 'name' -b 'bp' -pvalue 1.0 -output-dir 'output'*,

- RSEG: *rseg-diff -verbose -mode 3 -out 'output' -score 'score' -chrom 'chrom' -bin-size 'bp' -deadzones 'deadzonee' -duplicates 'sample1' 'sample2'*,
- ChIPComp: *ChIPComp(makeCountSet(conf,design,filetype="bam",species="hg19",binsize=bp))*,
- diffReps: *diffReps.pl -gname hg19 -report 'output' -treatment 'sample1' -control 'sample2' -btr 'control1' -bco 'control2' -window 'bp' -pval 1 -nsd 'marktype' -meth 'nb'*,
- DiffBind: *dba.report(dba.analyze(dba.contrast(dba.count(dba(sampleSheet = conf)), categories=DBA_CONDITION,minMembers=2))),th=1)*,

such that $\text{bp} = \{250, 500, 750, 100\}$ and $\text{marktype} = 'broad'$ if H3K27me3, H3K36me3, or EZH2, or $\text{marktype} = 'sharp'$ otherwise.

For DiffBind under 3 conditions (Figure 1, main text), the set of differential peaks included all peaks deemed to be differential by DiffBind under an FDR control of 0.05 simultaneously for all three pairwise contrast tests between the cell lines Helas3, Hepg2, and Huvec. In the particular genomic position shown in Figure 1, no differential peaks were reported by DiffBind.

For csaw, we used the following setup:

```
# List of bam files

bam.files = list.files(path=paste0(tmpdir, '/chip'), pattern='*.bam$', 
                       full.names=T)

bam.files


# Design matrix

design <- model.matrix(~factor(cell.type))

colnames(design) <- c("intercept", "cell.type")

design
```

```
# Parameters (PCR duplicates already removed and quality score filtered)
param <- readParam(dedup = F)

param

# Estimating the average fragment length (rescaling all to 200bp)
x = lapply(bam.files, correlateReads, param=param, max.dist=250)
multi.frag.lens = list(unlist(lapply(x, maximizeCcf)), 200)
multi.frag.lens

# Counting reads (for a window size of 250bp, for instance)
data <- windowCounts(bam.files, width = 250, ext = multi.frag.lens,
                      param = param, filter = 20)
data

# Filtering data
data.large <- windowCounts(bam.files, width=2500, bin=T, param=param)

bin.ab <- scaledAverage(data.large, scale=median(getWidths(data.large))/
                           median(getWidths(data)))

threshold <- median(bin.ab) + log2(2)

keep.global <- aveLogCPM(asDGEList(data)) > threshold

sum(keep.global)

# Creating filtered data
```

```

filtered.data <- data[keep.global,]

# Testing for DB (assuming composition bias is negligible,
# i.e. cell lines should exhibit a balanced number of DB regions)

y <- DGEList(assay(filtered.data), lib.size = filtered.data$totals)
y$samples$norm.factors <- 1
y$offset <- NULL
y <- estimateDisp(y, design)
fit <- glmQLFit(y, design, robust = TRUE)
out <- glmQLFTest(fit, contrast = contrast)
tabres <- topTags(out, nrow(out))$table
tabres <- tabres[order(as.integer(rownames(tabres))),]

merged <- mergeWindows(rowRanges(filtered.data), tol=tol,
                        max.width=max.width)
tabneg <- combineTests(merged$id, tabres)

```

For ChIPComp and DiffBind, candidate peaks were called in advance using MACS2 with the following syntax:

- MACS2: *macs2 callpeak -f BAM -g 2.80e+09 -B 'options' -t 'sample' -c 'control' -outdir 'output' -n 'filename'*

such that *options* = {–broad –broad-cutoff 0.1} if H3K27me3, H3K36me3, or EZH2, or *options* = {–q 0.01} otherwise.

In the chromatin segmentation analysis, ChromHMM was run with the commands:

- ChromHMM-BinarizeBed: *java -jar ~/ChromHMM/ChromHMM.jar BinarizeBam -b 500 -c 'controldir' /ChromHMM/CHROMSIZES/hg19.txt 'inputbamdir' 'input.txt'*

'outputbinarydir' and

- ChromHMM-LearnModel: *unset DISPLAY && java -jar ~/ChromHMM/ChromHMM.jar LearnModel -p 0 -b 500 'outputbinarydir' 'outputdir' 'numstates' hg19,*

such that 'controldir' is the directory with control experiments, 'inputbamdir' is the directory with input bam files, 'input.txt' is an experimental table, 'outputbinarydir' is the output directory with binarized files, 'outputdir' is the model output, and 'numstates' is the number of states in the model.

All ENCODE data analysis presented in the main text, and in this Web Appendix, were adjusted for input control samples in methods that are designed to do so. Specifically, differential peaks from ChIPComp, DiffBind, diffReps, and THOR were obtained on ChIP-seq samples while passing input control samples in their software call as an optional input argument argument. In contrast, input control samples were not considered in epigraHMM, cseg, and RSEG analyses.

A3 Software

epigraHMM was implemented in a R package that is available on the GitHub repository <https://github.com/plbaldoni/epigraHMM>.

epigraHMM is a package with a differential peak caller to detect differential enrichment regions from multiple ChIP-seq experiments with replicates. The main function of the package is *epigraHMM()*. The package allows the user to specify a set of parameters that control the Expectation-Maximization (EM) algorithm. These parameters include, for instance, the convergence (and termination) criteria of the algorithm and the threshold value for the rejection controlled EM algorithm. These parameters can be defined by the function *controlEM()*. Please refer to the package documentation (e.g. *?epigraHMM::epigraHMM*) for additional details and the complete help manual.

A4 Code

The necessary code to replicate the results presented in the main article and in the supplementary material can be downloaded from <https://github.com/plbaldoni/epigraHMMPaper>.

Supporting Information B

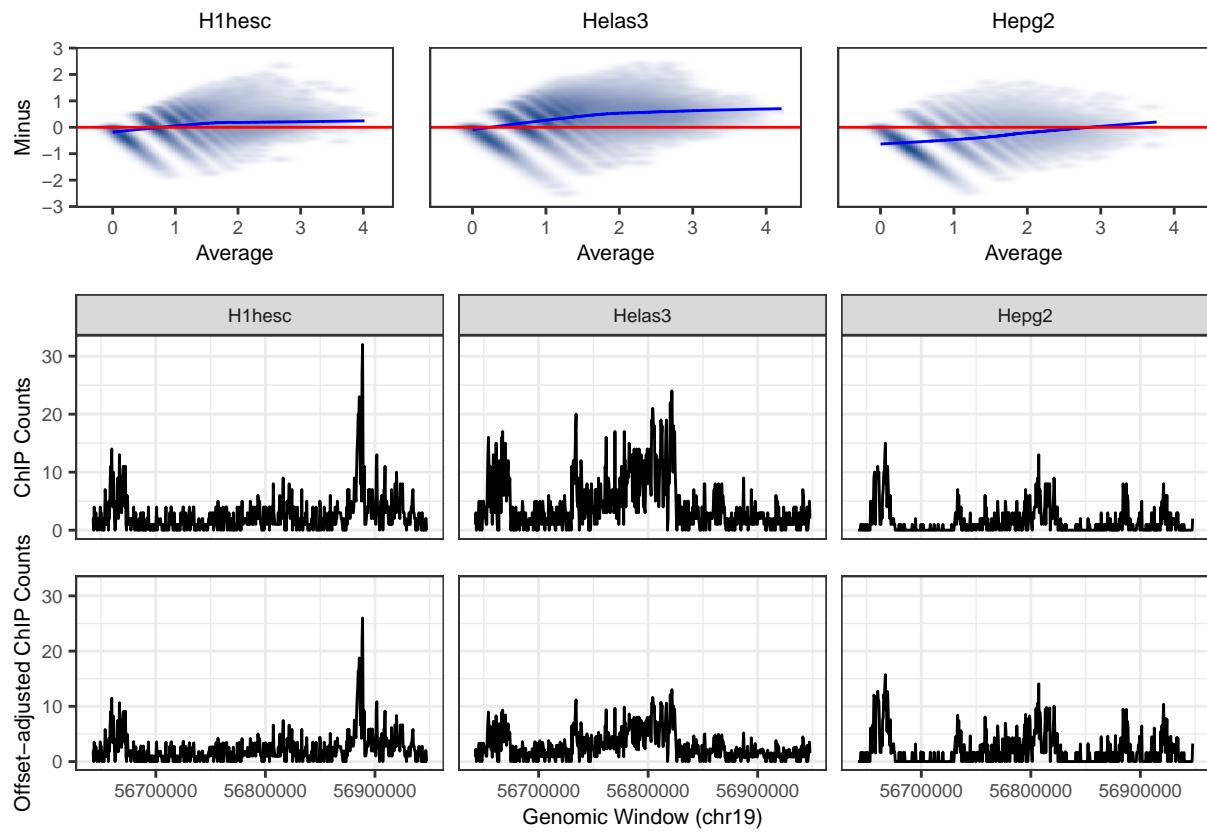
In this appendix, we present additional methodological results and an overview of the EM algorithm proposed in the main article.

B1 Adjustments for nuisance effects

B1.1 Normalization for non-linear biases via model offsets

In our analyses, we observed that the magnitude of the local differences in read counts between conditions changed with the average of local read coverage. Here, we accounted for these trended differences to avoid calling spurious differential peaks due to the different magnitude of library sizes across groups. Specifically, we implemented an approach similar to the non-linear normalization method used by csaw as follows [7]. First, we create a reference sample of read counts comprised by the geometric mean of read counts from all replicates and conditions. Then, we fitted a loess curve on the difference between the read counts of each sample and the reference, on the average of those two quantities. A similar approach was first implemented by [7] and is available in their software. Here, we add a continuity correction of 1 to avoid discarding genomic windows with zero counts. Using the smoothed curve as the model offset, we observed better results than a simple correction via either the total sum of read counts or cell-specific median log ratio. The rationale behind this approach is to create a reference library in which each genomic window is the geometric mean of counts across

all conditions and replicates, and then read counts are properly adjusted by accounting for the smoothed differences between each individual library and the reference library. A useful way to evaluate the performance of this normalization method is to compare samples with respect to their adjusted read counts. For example, plotting the ratios between counts and the calculated offsets $y_{hij} / \exp(u_{hij})$ for all samples in the study. In Figure 1 we show an example of a genomic region from three analyzed cell lines and their respective MA plot, unadjusted ChIP counts, and offset-adjusted ChIP counts. After accounting for the offset, the read counts from Helas3 are adjusted to its larger library size with respect to the other under sequenced cell lines.



Web Figure 1: MA plot of read counts from three distinct analyzed cell lines (top), unadjusted ChIP read counts (center), and offset-adjusted ChIP read counts (bottom) from a given genomic region on chromosome 19. The blue line in the MA plots shows the offset created via loess smoothing.

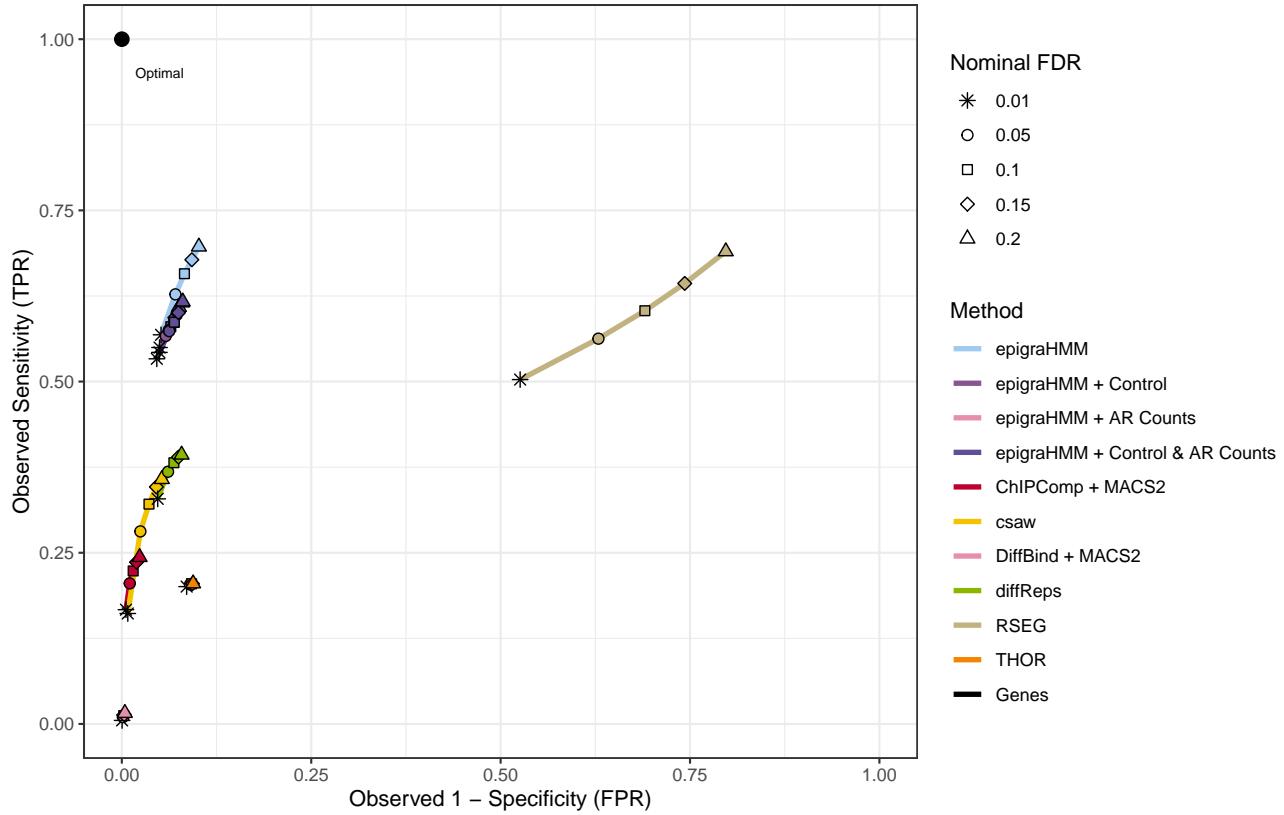
B1.2 Input control adjustment in differential peak calling

Our implementation (Web Appendix A4) allows the optional inclusion of continuous covariates in the model with state-specific parametrization. The main purpose of the inclusion of such covariates in the model is the adjustment for input control (or any other continuous variable, such as autoregressive counts) that can be helpful in distinguishing background from enrichment signal. Several methods for differential peak calling allow the inclusion of input control in their computational framework [13, 11, 2, 1]. However, [7] point out that ”(...) controls are mostly irrelevant when testing for DB between ChIP samples.”. To evaluate this claim, we ran an analysis of real data and simulated data while accounting for the input control effect.

To asses whether accounting for input control effect leads to an improvement in performance, we utilized the smoothing technique proposed by [2] to account for input controls and autoregressive counts. Specifically, we fitted generalized additive models (GAM, instead of loess smoothing) in the data normalization step while accounting for input control (or autoregressive counts) as a covariate. The resulting fitted curve was then used in the analysis as model offsets.

First, we analyzed real data by smoothing the input control effect and autoregressive counts with a two-step approach. Specifically, we first called peaks without the inclusion of extra covariates in the model, and then utilized the called differential peaks from the first step to smooth the covariates for each HMM predicted state. Predicted smoothing curves from the GAM approach were then passed as model offsets in a second step of analysis. As claimed by [7], we observed minor differences in the results that would justify their inclusion in the analysis. Results from the histone modification mark H3K36me3 are presented in Figure 2.

Next, we reasoned that our approach of modeling input control effect with state-specific parametrization could not be ideal, since independent controls were available for every sample and there could exist sample-specific effects not captured by our model. We then at-

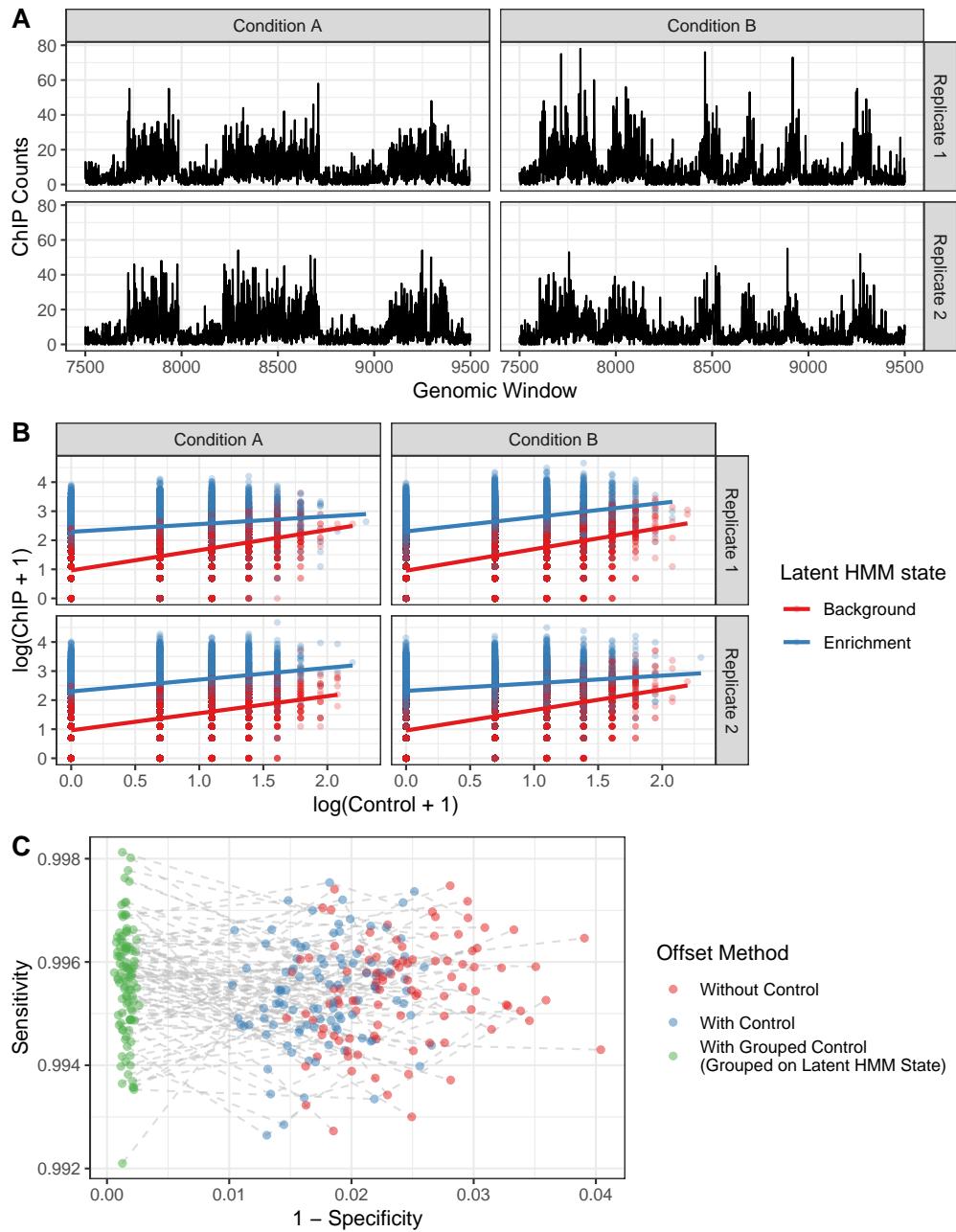


Web Figure 2: ROC curves for H3K36me3 utilizing no input controls (epigraHMM), input control only (epigraHMM + Control), autoregressive counts only (epigraHMM + AR Counts), and smoothing of both input controls and autoregressive counts (epigraHMM + Control & AR Counts)

tempted to verify the utility of including input control into the differential binding analysis by simulating data where ChIP-counts were generated such that their log-mean had a linear relationship with input controls (Figure 3). We then fitted three different models that differed regarding the inclusion of input control: a model without control, a model with control, and a model with controls where the smoothing was calculated separately for each latent HMM state. Again, results did not show significant improvement by including the effect of control in the analysis.

Overall, we did not observe a significant improvement in performance by including input control in differential peak calling. Although several methods do offer the option of including

controls in their analysis pipeline, we did not find that their inclusion was justifiable under our modeling assumptions. Our findings are in agreement with [7].



Web Figure 3: Results from simulated data (A) where the log-means of ChIP-seq counts were generated as a linear function of input controls (B). Sensitivity/specificity analyses did not show significant improvement by including the effect of control in the offset scheme.

B1.3 GC-content bias in broad marks

We performed a simulation study to assess the benefits and ability of epigraHMM to adjust for GC-content bias. In this study, we borrowed the method from [15] and used their normalizing denominators to adjust for GC-content bias in our simulations study. As [15] suggest, we utilized their normalizing offsets into our own model to account for the bias results from the simulated GC content. Because the simulated data varied with respect to the sequencing depth, we further included in our model normalizing offsets to adjust for the excess of read enrichment using the loess normalization approach as described in Section B1.1.

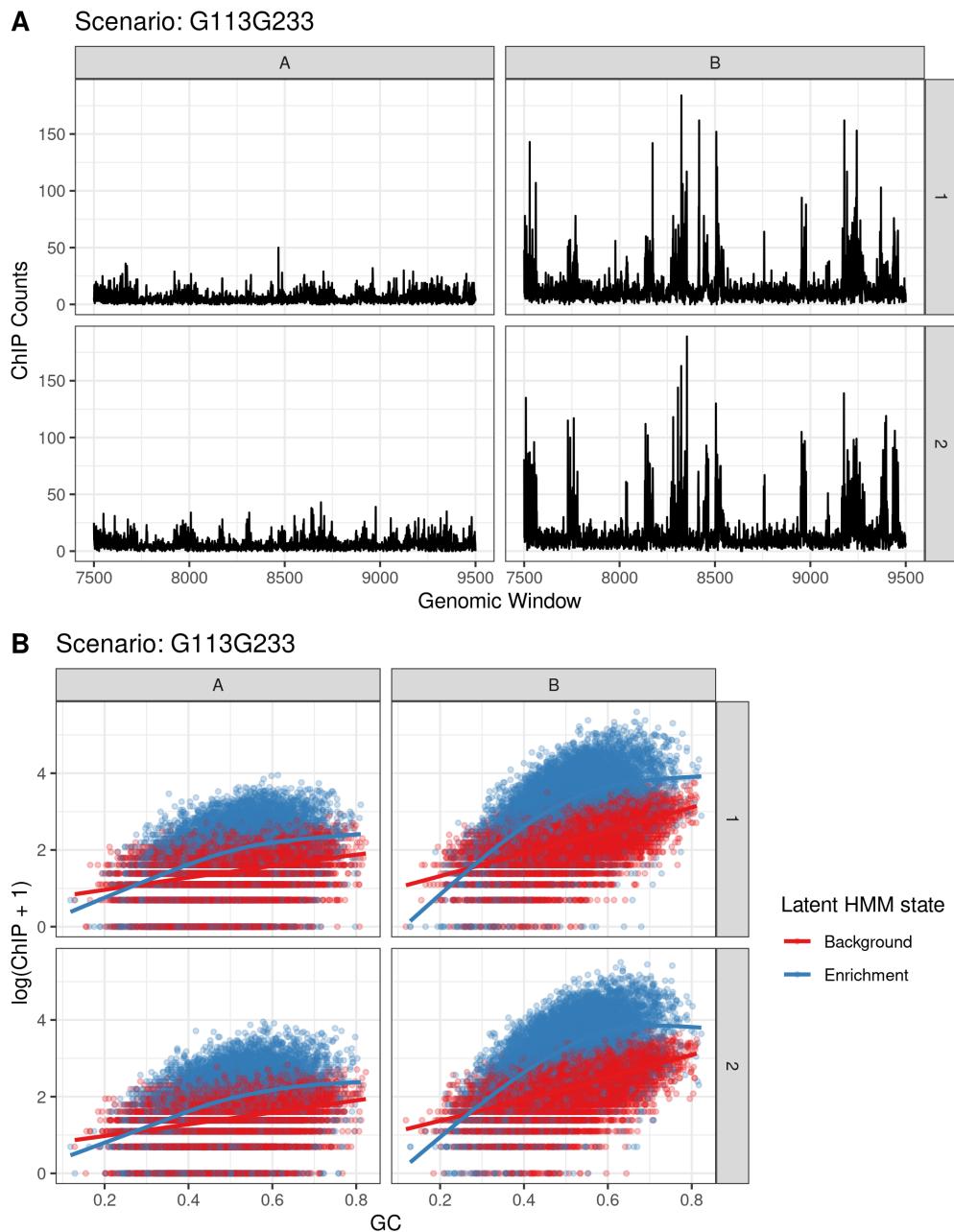
Data was generated as described in Section 4.1 of our paper, with the addition that GC-content bias was introduced to the mean model. As suggested by [15], the simulated GC-content effect varied from condition to condition as well as the sequencing depth. In this simulation study, we assessed decreasing, neutral, and increasing effects of GC-content on the mean read count distribution. In Figure 4 we show an example of simulated data with increasing GC-content effect on ChIP-seq counts in both conditions. In this example, these conditions varied regarding the depth of sequencing reads.

We assessed the model performance while adjusting for sequencing depth only, GC-content effect only, and simultaneously adjusting for both GC-content and sequencing depth. We observed that the optimal results were achieve only after normalization for both GC and sequencing depth bias. Despite the improvement in performance after the non-linear normalization, we were able to achieve optimal results in terms of true positive rate and false discovery rate of peak calls only after accounting for GC-content bias. These results not only show that our model is able to account for such type of bias, here computed using the methodology from [15], but also highlight the importance of such an adjustment. These results are shown in Figure 5.

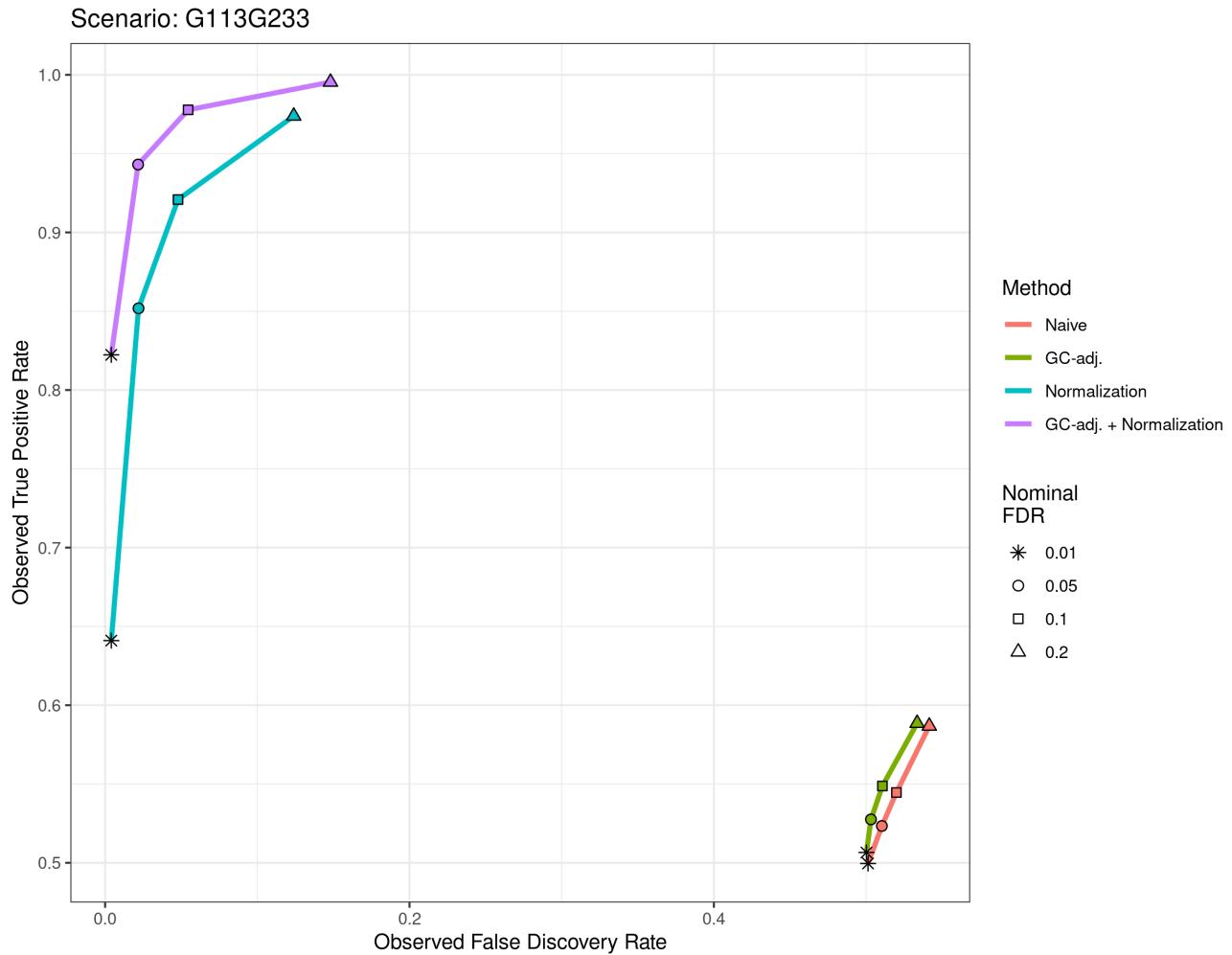
To better visualize the results on a per data set basis, Figure 6 shows the sensitivity and 1-specificity results from all 100 simulated datasets for the aforementioned example. Here, we see that for all data sets only after account for GC-content bias we were able to achieve

optimal results.

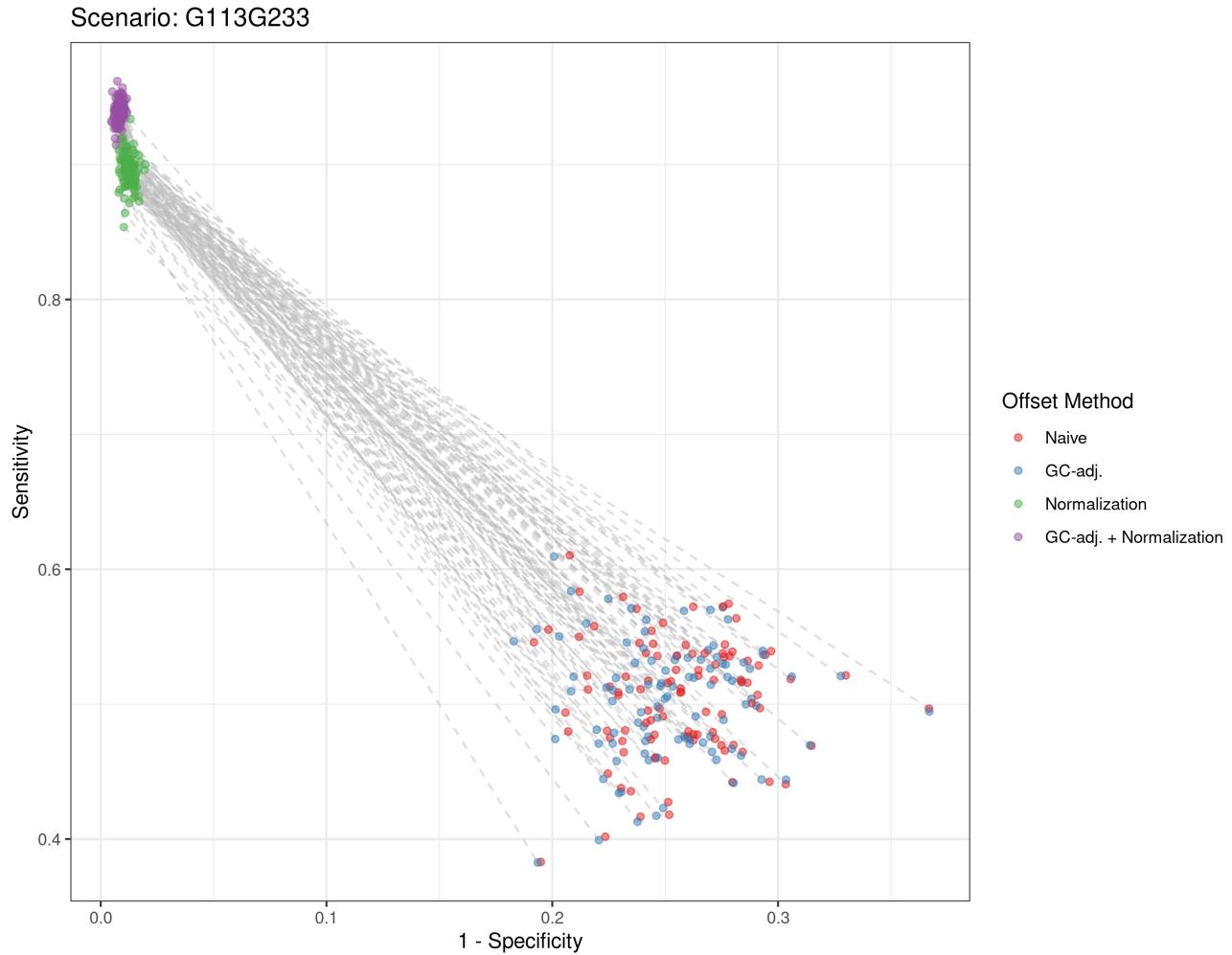
In Figure 7 below, we present smoothed scatter plots of log-transformed ChIP-seq counts (y-axis) and the respective GC content (x-axis) for the ENCODE EZH2 data that we analyzed. While using a 500bp window size to compute read counts, we did not observe enough evidence to justify account for such an effect in our analyses. Note that, when using such read count resolution, the effect of GC content can be hardly seen. In addition, there is no evident separation of differential GC-content effect for peak and background regions, as reported in [15].



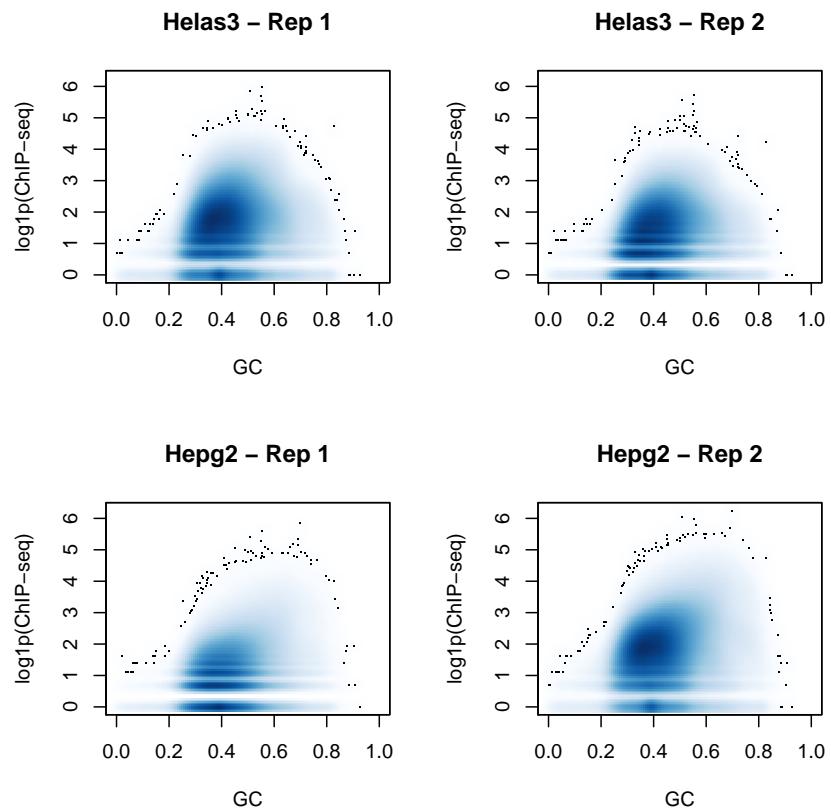
Web Figure 4: Simulation study for GC-content bias correction with epgraHMM. In (A), example of a genomic region from the simulated data. In (B), simulated GC content effect on ChIP-seq counts. Scenario: low depth in condition 1, high depth in condition 2; positive GC effect on ChIP-seq counts in both conditions.



Web Figure 5: Simulation study for GC-content bias correction with epigraHMM. Average observed true positive rate (y-axis) and observed false discovery rate (x-axis) in 100 simulated data sets for four different GC-content bias correction strategies: naive model without normalization for sequencing depth or GC-content bias, GC-content adjusted model, sequencing depth adjusted model, and GC-content and sequencing depth adjusted model. Optimal results are obtained only when the effect of GC-content is considered together with normalization for depth. The GC-content bias was accounted via model offsets in epigraHMM using the normalizing denominators from [15].



Web Figure 6: Simulation study for GC-content bias correction with epgraHMM. Observed sensitivity (y-axis) and observed 1-specificity (x-axis) for each of the 100 simulated data sets four different GC-content bias correction strategies: naive model without normalization for sequencing depth or GC-content bias, GC-content adjusted model, sequencing depth adjusted model, and GC-content and sequencing depth adjusted model. Optimal results are obtained only when the effect of GC-content is considered together with normalization for depth. The GC-content bias was accounted via model offsets in epgraHMM using the normalizing denominators from [15]. Dashed lines connect the observed metrics for each simulated dataset.



Web Figure 7: Smoothed scatter plots of log-transformed EZH2 ENCODE ChIP-seq counts and GC content.

B2 Bayesian Information Criterion (BIC) for Hidden Markov Models

The BIC for hidden Markov models has been discussed by [16]. For the presented three-state HMM, one can calculate the BIC as

$$BIC = -2 \log \left(\sum_{r=1}^3 f_{Mr}^p \right) + (11 + L) \log \left(M \sum_{h=1}^G n_h \right), \quad (\text{B.1})$$

where f_{Mr}^p is the forward probability pertaining to the r^{th} state calculated at the (last) M^{th} genomic window (as detailed in the Appendix of the main text), L is the number of mixture components, G is the number of conditions, and n_h is the number of replicates pertaining to condition h . The number of model parameters to be estimated is $(11 + L)$: 6 transition probabilities, 2 initial probabilities, 4 model coefficients pertaining to the emission distributions, and $L - 1$ prior probabilities from the mixture model.

As shown in the main text, the proposed HMM is robust to situations where certain combinatorial patterns are rare. However, if pruning rare combinatorial patterns is still of interest, such a task can be performed by making use of the BIC. For the analysis of G experimental conditions with a given BIC threshold ϵ , say $\epsilon = 0.01$, and $L = 2^G - 2$ mixture components, one can prune rare combinatorial patterns by the following algorithm.

1. Fit the three-state HMM with L mixture components (model L) and compute the model BIC, BIC_L , as in Equation B.1.
2. Fit a reduced three-state HMM with $L - 1$ mixture components (model $L - 1$) by excluding the component associated with the rarest combinatorial pattern of enrichment. Compute its BIC, BIC_{L-1} .
3. Calculate $\Delta BIC = (BIC_{L-1} - BIC_L)/BIC_L$. If $|\Delta BIC| \leq \epsilon$, set $L \leftarrow L - 1$ and return to step 1. If $|\Delta BIC| > \epsilon$, stop and set the model L as the final model.

In scenarios where the number of mixture components is smaller than $2^G - 2$, the implemented method initializes the EM algorithm by clustering genomic windows with respect to the posterior probabilities of enrichment obtained from a initial run of a two-state HMM to classify genomic windows into background and enrichment windows. Such an initialization improves the overall computation time by reducing the time to convergence of the presented EM algorithm.

We also compare the results of our method with ChromHMM, an algorithm developed for chromatin segmentation. In Figure 8, panels A-D, we present results of ChromHMM with 3 (ideal), 4, 5, and 6 states. By using the BIC for model selection, one could easily choose the number of biologically relevant mixture components to be included in the model, a task that may not be as straightforward in methods such as ChromHMM (see Supplementary Figure 4 in [3]). Our method offers the benefit of simultaneously detecting differential peaks and classifying the combinatorial pattern of enrichment through mixture model posterior probabilities even in the context of genomic segmentation. Despite the choice of the number of mixture components via BIC, epgraHMM appeared to be robust in scenarios with rare combinatorial patterns. This is so because we utilize a constrained parametrization in which only 4 GLM-specific parameters need to be estimated regardless of the number of mixture components, and in the genome-wide analysis of ChIP-seq data one often has enough data to estimate such quantities.

B2.1 Simulation Study

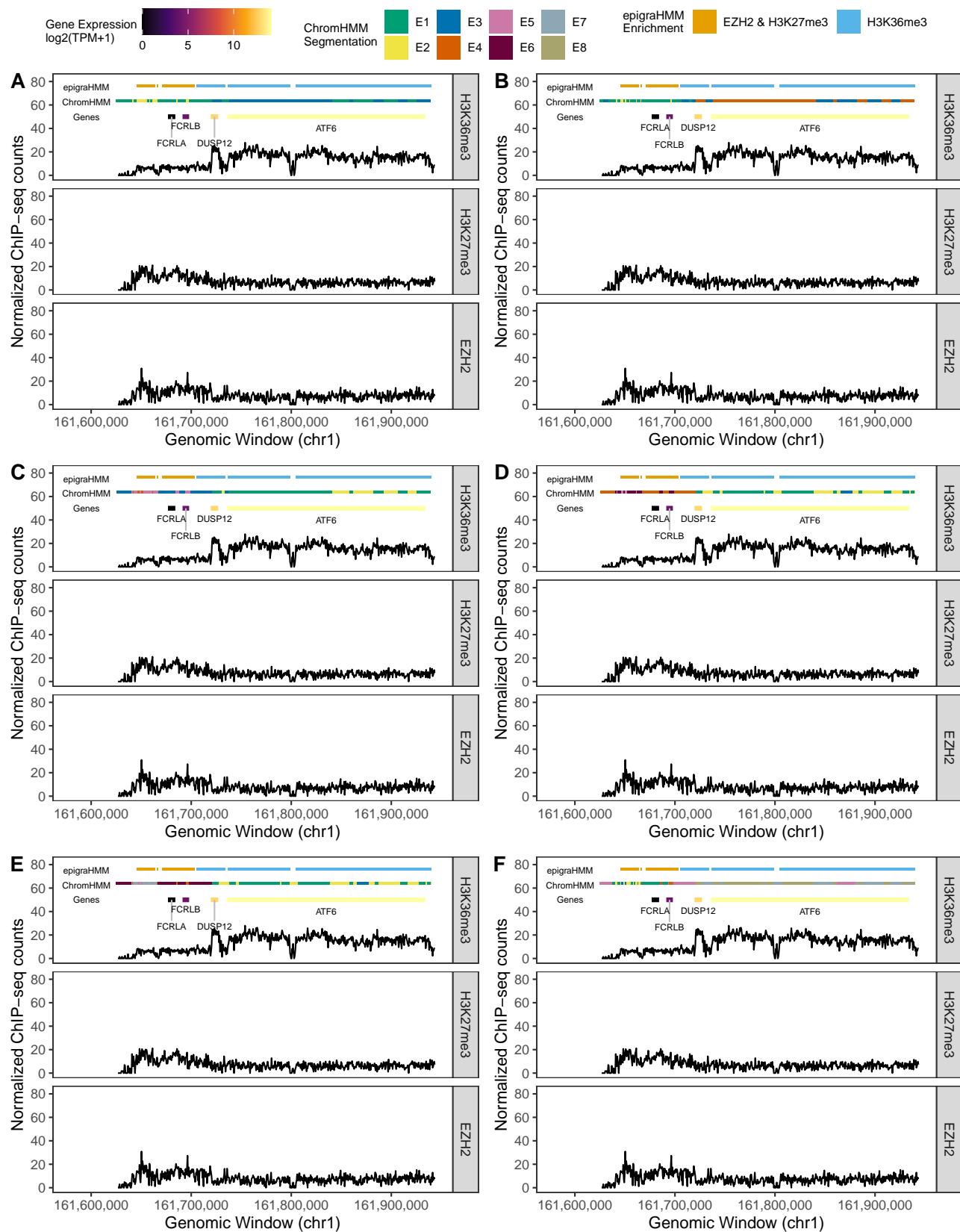
We assessed the performance of our model selection scheme via BIC with a simulation study. To this end, we re-generated data in the same fashion as described in Section 4 of our paper while removing certain combinatorial patterns. Specifically, to mimic the ENCODE data analyzed in Section 5 of our paper, we generated data for which 3 conditions were available but only 2 out of 6 possible differential combinatorial patterns were available.

First, it is important to note that for an analysis of 3 conditions, there are $2^3 - 2 = 6$

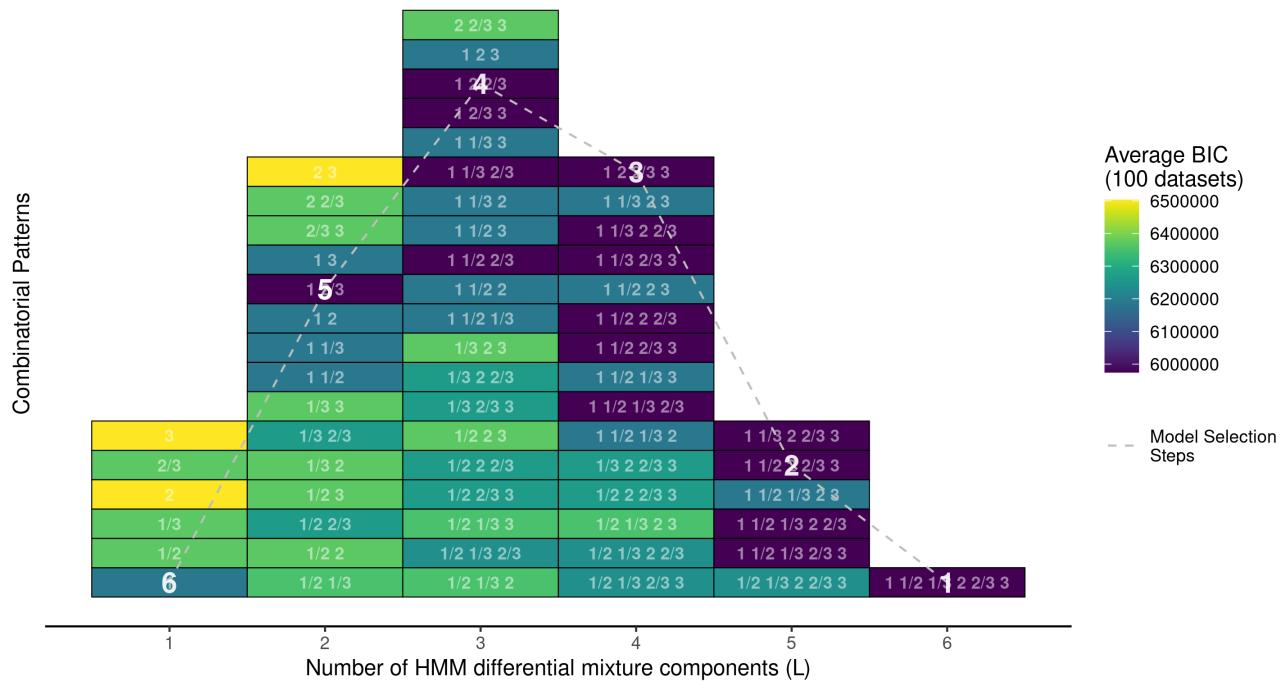
possible combinatorial patterns of differential enrichment. In this context, for a mixture model with L components, there are $\binom{6}{L}$ possible ways to model L components. The choice of the number L of components therefore needs to reflect the relevant biological patterns in the data. In this study, we assessed the algorithm presented in Section B2 by fitting all $\sum_{l=1}^6 \binom{6}{l}$ possible model and computing their BIC for 100 simulated data sets. On average, the optimal chosen model should be the one with $L = 2$ components *and* with the components modeling those relevant ones. In this study, the two relevant patterns simulated were enrichment in condition 1 alone, and co-enrichment of conditions 2 and 3 (similar to Figure 6 of the main paper). As in Section 4 of our paper, we assessed the performance of our model under different SNR levels, different number of replicates per condition, and different genome sizes.

We present our results in Figures 9, 10, and 11. We start our model selection by fitting the full model with 6 components ($L=6$, rightmost block of the triangle with the highlighted '1'). We then proceed with our algorithm as described in this section and remove 1 combinatorial pattern at a time until the difference in BIC is large enough. As we can see from Figures 9, 10, and 11, this is achieved when $L=2$ and the combinatorial patterns being modeled is 1 (enrichment condition 1) and 2/3 (co-enrichment of condition 2 and 3). As we can see, our model selection scheme correctly selected the optimal model with the proper number of combinatorial patterns L and the proper patterns themselves.

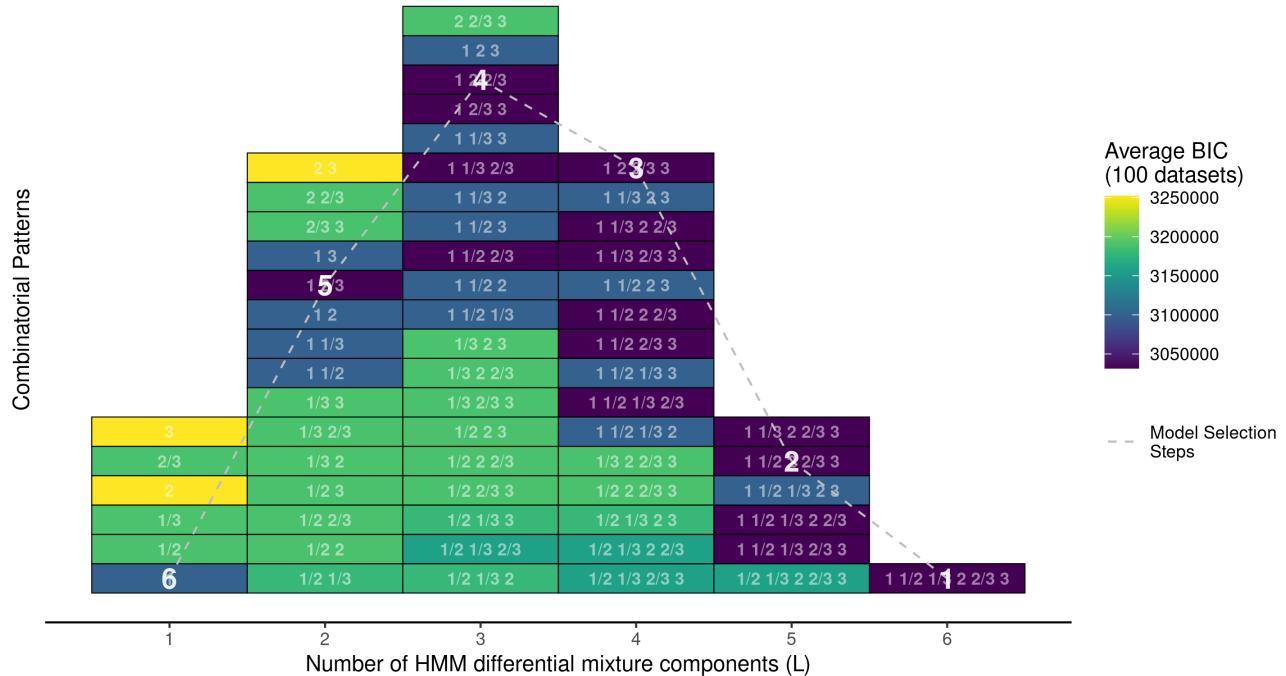
In accordance with the results presented in Section 4 of our paper, we note that the number of replicates played a major role in the success of our model selections scheme. For example, by comparing Figure 9 and Figure 11, we see that more technical/biological replicates lead to models that can be better distinguished in terms of BIC when one is searching for the optimal patterns and number of combinatorial patterns. In summary, the presented model selection scheme allows users to prune combinatorial patterns of interest when one is not interested in modeling certain rare biological patterns from the data.



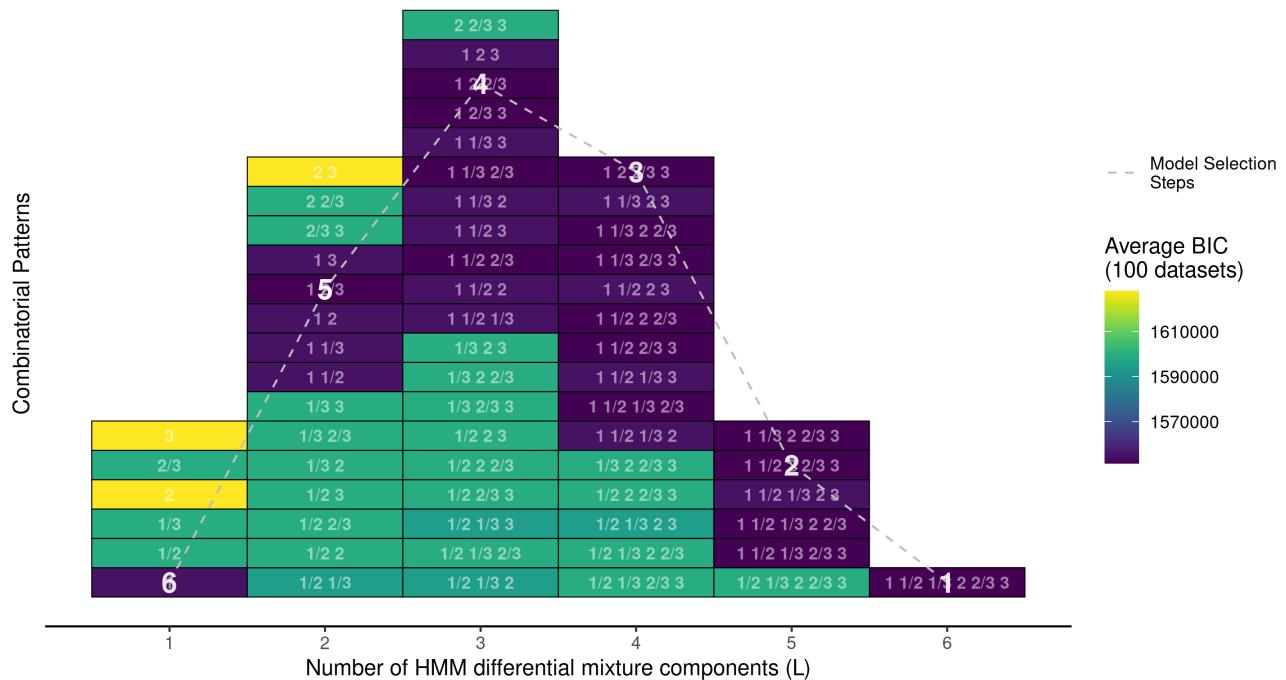
Web Figure 8: Comparative analysis of epigraHMM and ChromHMM with 3 (ideal), 4, 5, 6, 7, and 8 states (panels A-F).



Web Figure 9: Simulation study for model selection via BIC. Simulation scenario: H3K36me3 data, 3 conditions, 4 replicates per condition, 10^5 genomic windows, observed (estimated from ENCODE) SNR. Dashed lines indicate order of model selection scheme beginning with the full model ('1') and ending with the reduced model ('6'). The optimal model is the one with highlighted '2', since it had a significant lower BIC than any other model with a single differential component. Highlighted number in each block indicated the combinatorial pattern being modeled. For instance '1/2 2 3' indicates a model with three components and the following patterns: co-enrichment of condition 1 and 2, enrichment of condition 2 alone, and enrichment of condition 3 alone.



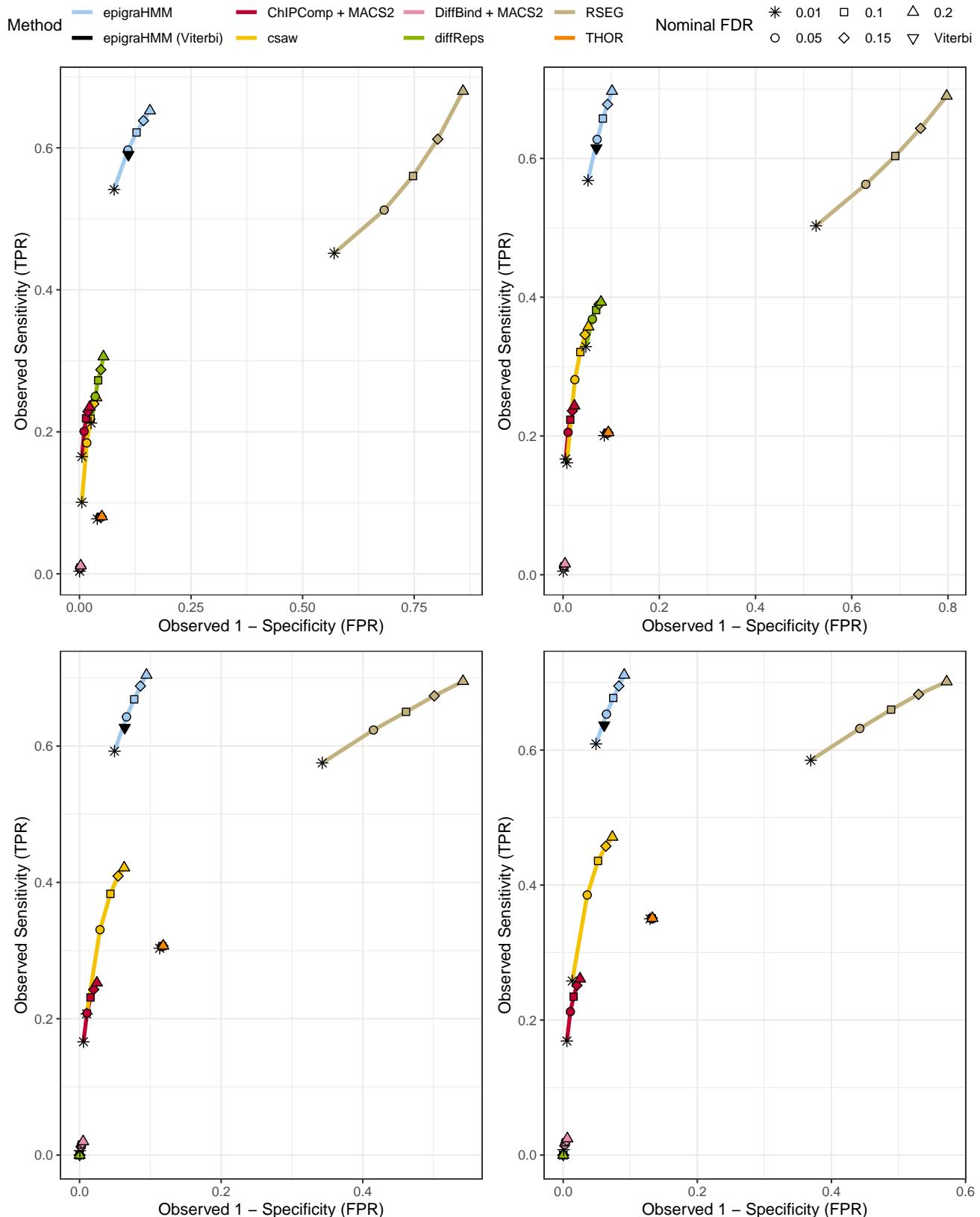
Web Figure 10: Simulation study for model selection via BIC. Simulation scenario: H3K36me3 data, 3 conditions, 2 replicates per condition, 10^5 genomic windows, observed (estimated from ENCODE) SNR. Dashed lines indicate order of model selection scheme beginning with the full model ('1') and ending with the reduced model ('6'). The optimal model is the one with highlighted '2', since it had a significant lower BIC than any other model with a single differential component. Highlighted number in each block indicated the combinatorial pattern being modeled. For instance '1/2 2 3' indicates a model with three components and the following patterns: co-enrichment of condition 1 and 2, enrichment of condition 2 alone, and enrichment of condition 3 alone.



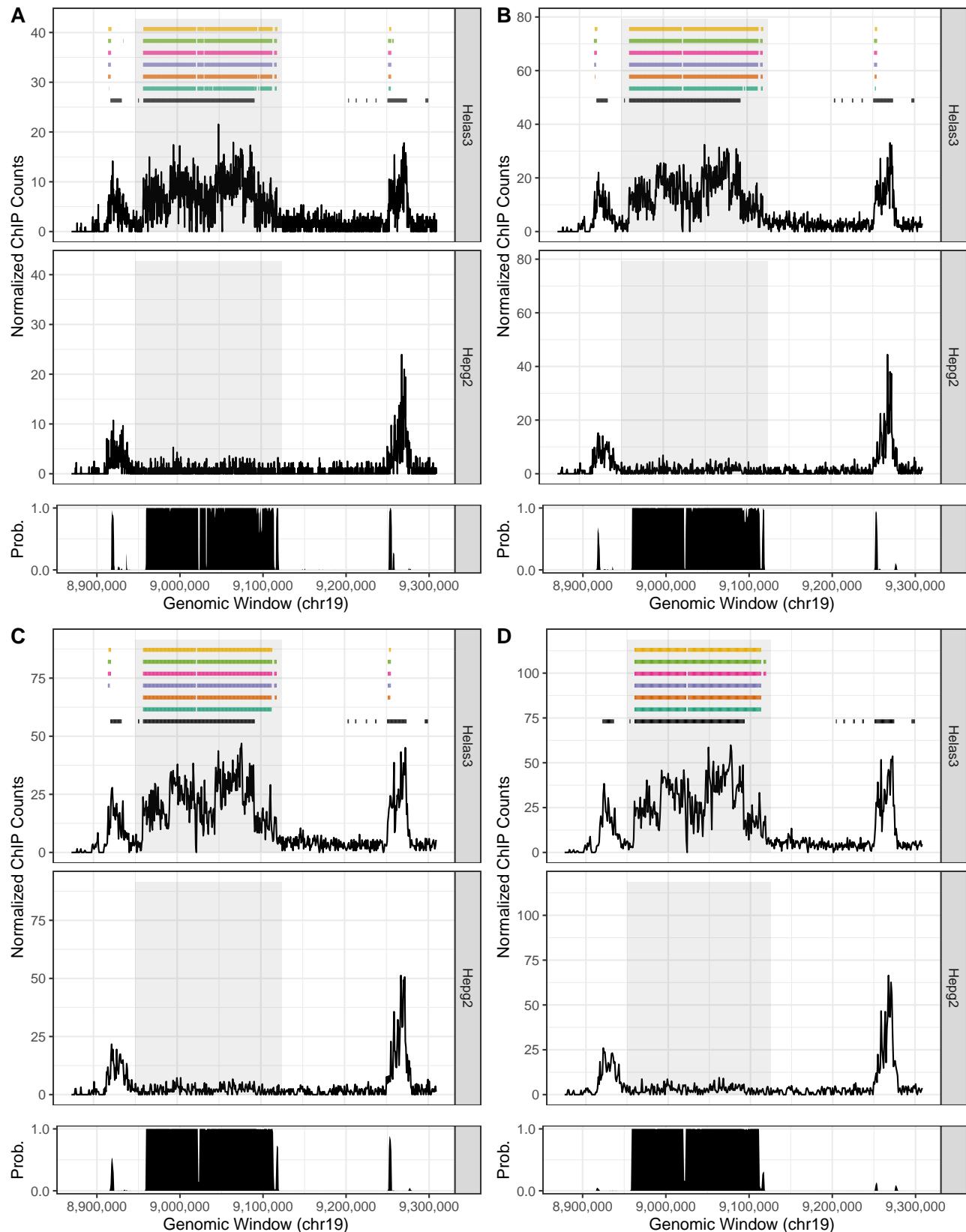
Web Figure 11: Simulation study for model selection via BIC. Simulation scenario: H3K36me3 data, 3 conditions, 1 replicate per condition, 10^5 genomic windows, observed (estimated from ENCODE) SNR. Dashed lines indicate order of model selection scheme beginning with the full model ('1') and ending with the reduced model ('6'). The optimal model is the one with highlighted '2', since it had a significant lower BIC than any other model with a single differential component. Highlighted number in each block indicated the combinatorial pattern being modeled. For instance '1/2 2 3' indicates a model with three components and the following patterns: co-enrichment of condition 1 and 2, enrichment of condition 2 alone, and enrichment of condition 3 alone.

B3 Study on FDR Control and the Viterbi Algorithm

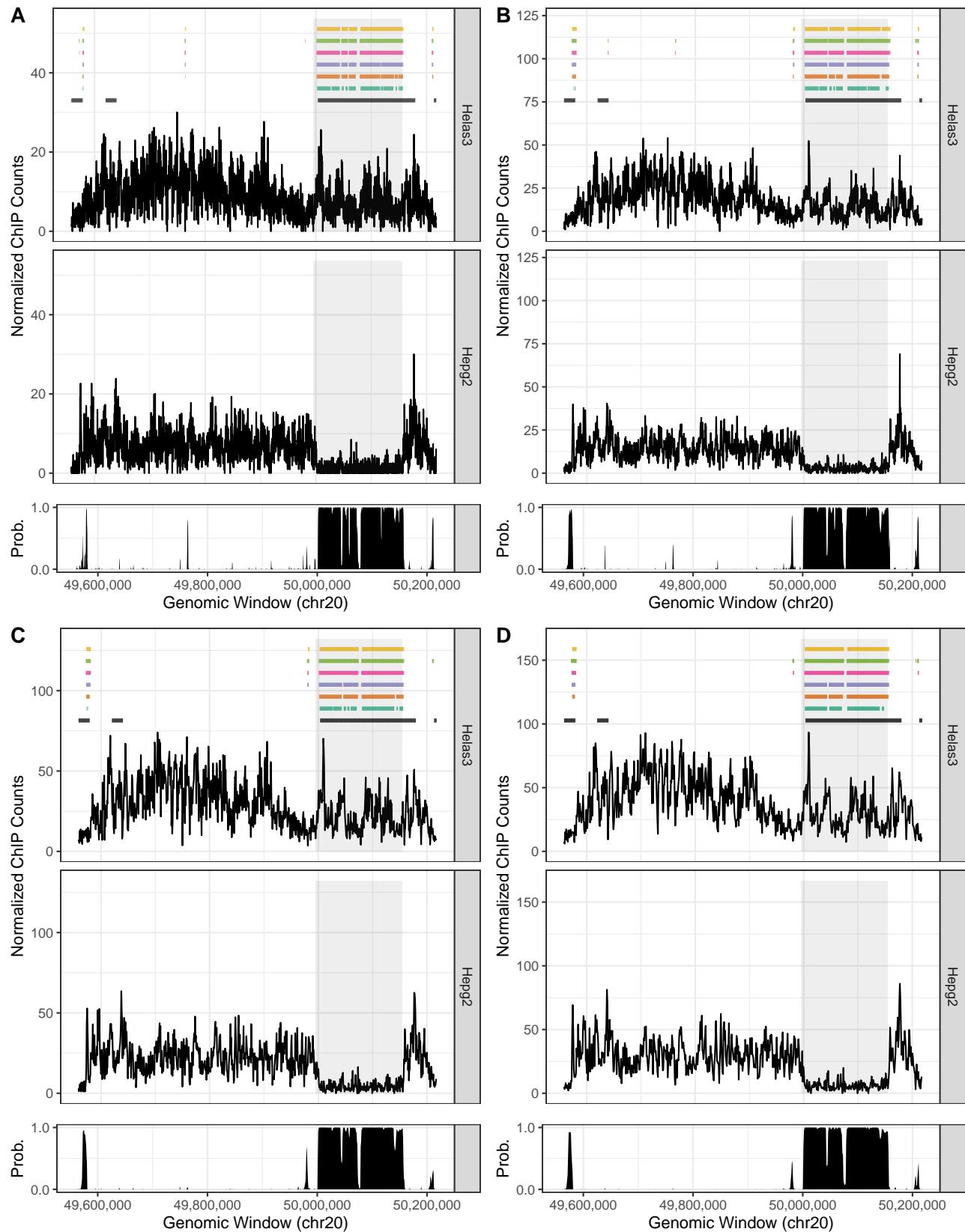
In this section, we present results comparing the FDR controlling approach with the Viterbi algorithm. We evaluated the FDR approach using cutoffs 0.01, 0.05, 0.10, 0.15, and 0.20. We compared the results between the two approaches using window sizes of 250bp, 500bp, 750bp, and 1000bp. Overall, we observed that the Viterbi sequence of states led to similar results than the sequences based on FDR control cutoffs across all choices of window size. Specifically, we observed that the sensitivity and specificity of the sequence of Viterbi states were close to those from FDR control, in particular for FDR control 0.10. These results are shown in Figure 12. These facts are also reflected by the length and number of called peaks. In Figures 13 and 14 we show examples of peak calls of H3K26me3 and H3K27me3, respectively, from all FDR control cutoffs and the Viterbi sequence of states. Overall, we observe minor differences regarding the size of peak calls of the Viterbi and FDR control sequences across different choices of window sizes. These differences were mainly present in the data for H3K27me3, which is known to be a histone mark that expands through broader domains than H3K36me3. Finally, it is worth noting that the Viterbi algorithm gives us a way to call peaks that does not depend on the choice of the FDR cutoff.



Web Figure 12: FDR-based results from broad marks and Viterbi-based result from epigraHMM with 250bp (A), 500bp (B), 750bp (C), and 1000bp (D).



Web Figure 13: FDR- and Viterbi-based peak calls from H3K36me3 with 250bp (A), 500bp, (B), 750bp (C), and 1000bp (D).



Web Figure 14: FDR- and Viterbi-based peak calls from H3K27me3 with 250bp (A), 500bp, (B), 750bp (C), and 1000bp (D).

B4 HMM Emission Distributions

For the consensus background ($r = 1$) and consensus enrichment ($r = 3$) states, the emission distribution function is

$$\begin{aligned} f_r(\mathbf{y}_{..j} | \boldsymbol{\psi}_r) &= \prod_{h=1}^G \prod_{i=1}^{n_h} f_r(y_{hij} | \boldsymbol{\psi}_r), \quad r \in \{1, 3\} \quad \text{and} \quad y_{hij} \in \{0, 1, 2, \dots\}, \\ &= \prod_{h=1}^G \prod_{i=1}^{n_h} \Pr(Y_{hij} = y_{hij} | Z_j = r; \boldsymbol{\psi}_r), \\ &= \prod_{h=1}^G \prod_{i=1}^{n_h} \frac{\Gamma(y_{hij} + \phi_r)}{y_{hij}! \Gamma(\phi_r)} \left(\frac{\phi_r}{\mu_{(r,hij)} + \phi_r} \right)^{\phi_r} \left(\frac{\mu_{(r,hij)}}{\mu_{(r,hij)} + \phi_r} \right)^{y_{hij}}. \end{aligned} \quad (\text{B.2})$$

For the differential state ($r = 2$), the emission distribution is

$$\begin{aligned} f_2(\mathbf{y}_{..j} | \mathbf{x}; \boldsymbol{\delta}, \boldsymbol{\psi}_2) &= \sum_{l=1}^L \delta_l f_{(2,l)}(\mathbf{y}_{..j} | \mathbf{x}_l; \boldsymbol{\psi}_{(2,l)}), \quad y_{hij} \in \{0, 1, 2, \dots\}, \\ &= \sum_{l=1}^L \delta_l \prod_{h=1}^G \prod_{i=1}^{n_h} Pr(Y_{hij} = y_{hij} | Z_j = 2, \mathbf{x}_l; \boldsymbol{\psi}_{(2,l)}), \\ &= \sum_{l=1}^L \delta_l \prod_{h=1}^G \prod_{i=1}^{n_h} \frac{\Gamma(y_{hij} + \phi_{(2,l,h)})}{y_{hij}! \Gamma(\phi_{(2,l,h)})} \left(\frac{\phi_{(2,l,h)}}{\mu_{(2,l,hij)} + \phi_{(2,l,h)}} \right)^{\phi_{(2,l,h)}} \times \\ &\quad \times \left(\frac{\mu_{(2,l,hij)}}{\mu_{(2,l,hij)} + \phi_{(2,l,h)}} \right)^{y_{hij}}. \end{aligned} \quad (\text{B.3})$$

Apart from the offset u_{hij} , we will assume that replicates from the same (different) condition share common (distinct) mean and dispersion parameters under every mixing probability distribution $f_{(2,l)}$. To define all possible combinations of background and enrichment across G conditions, we consider the following sets of singletons A_1 , pairs A_2, \dots , and $(G-1)$ -tuples

A_{G-1} such that

$$\begin{aligned} A_1 &= \left\{ a^{(1)} \mid a^{(1)} \in \mathbb{G}_+ \text{ and } a^{(1)} \leq G \right\}, \\ A_2 &= \left\{ (a_1^{(2)}, a_2^{(2)}) \mid (a_1^{(2)}, a_2^{(2)}) \in \mathbb{G}_+^2 \text{ and } a_1^{(2)} < a_2^{(2)} \leq G \right\}, \\ &\vdots \\ A_{G-1} &= \left\{ (a_g^{(G-1)})_{g=1}^{G-1} \mid (a_g^{(G-1)})_{g=1}^{G-1} \in \mathbb{G}_+^{G-1} \text{ and } a_1^{(G-1)} < \dots < a_{G-1}^{(G-1)} \leq G \right\}. \end{aligned}$$

The union of all sets $A = \cup_{k=1}^{G-1} A_k$ contains an exhaustive list of $L = 2^G - 2$ elements that determines the differential pattern across G conditions such that each element of A indicates which of the G conditions are enriched. For instance, if $G = 3$, $A_1 = \{1, 2, 3\}$ and $A_2 = \{(1, 2), (1, 3), (2, 3)\}$ define the six possible combinations of enrichment and background across three conditions. Then, we define a bijective mapping $A \rightarrow S_1, \dots, S_L$ and let $x_{hl} = I(h \in S_l)$ indicate whether the read count of genomic window j from replicate i of condition h is enriched in the mixture component l . We model the log-mean $\mu_{(2,l,hij)}$ and log-dispersion $\phi_{(2,l,h)}$ of mixture l from the emission distribution (B.3) as

$$\begin{aligned} \log(\mu_{(2,l,hij)}) &= \beta_1 + \beta_3 x_{hl} + u_{hij}, \quad \text{and} \\ \log(\phi_{(2,l,h)}) &= \lambda_1 + \lambda_3 x_{hl}. \end{aligned}$$

According to this parametrization, β_1 and λ_1 are the baseline log-mean and log-dispersion parameters of the read count distribution from replicates of conditions that are not enriched under the mixing distribution l . Conversely, $\beta_1 + \beta_3$ and $\lambda_1 + \lambda_3$ are the baseline log-mean and log-dispersion parameters of the read count distribution from replicates of conditions enriched under the mixing distribution l . This choice of parametrization ensures that windows exhibiting differential enrichment across conditions share means and dispersions that are common between the remaining non differential HMM states.

A pseudo code of the presented EM algorithm is below.

1. Initialize $\boldsymbol{\pi}^{(0)}, \boldsymbol{\gamma}^{(0)}, \boldsymbol{\delta}^{(0)}, \beta_1^{(0)}, \beta_3^{(0)}, \lambda_1^{(0)}, \lambda_3^{(0)}$, such that $\sum_{r=1}^3 \pi_r^{(0)} = 1$ and $\sum_{s=1}^3 \gamma_{rs} = 1$.
2. E step ($t \geq 1$),
 - (a) Calculate $Pr(Z_j = r | \mathbf{y}, \mathbf{x}; \boldsymbol{\Psi}^{(t-1)})$ and $Pr(Z_{j-1} = r, Z_j = s | \mathbf{y}, \mathbf{x}; \boldsymbol{\Psi}^{(t-1)})$ for all r and s in $\{1, 2, 3\}$ and $j = 1, \dots, M$ via Forward-Backward algorithm as detailed in Appendix B of the main article
 - (b) Calculate $Pr(W_{jl} = 1 | Z_j = 2, \mathbf{y}_{..j}, \mathbf{x}; \boldsymbol{\Psi}^{(t-1)})$ for all $l \in \{1, \dots, L\}$ and $j = 1, \dots, M$ as $f_{(2,l)}(\mathbf{y}_{..j} | \mathbf{x}_l; \boldsymbol{\psi}_{(2,l)}^{(t-1)}) \delta_l^{(t-1)} / \sum_{k=1}^L f_{(2,k)}(\mathbf{y}_{..j} | \mathbf{x}_k; \boldsymbol{\psi}_{(2,k)}^{(t-1)}) \delta_k^{(t-1)}$
3. M step ($t \geq 1$),
 - (a) Maximize Equation (5) (main text) with respect to the initial and transition probabilities to obtain for all r and s in $\{1, 2, 3\}$

$$\pi_r^{(t)} = Pr(Z_1 = r | \mathbf{y}, \mathbf{x}; \boldsymbol{\Psi}^{(t-1)})$$

$$\gamma_{rs}^{(t)} = \sum_{j=2}^M Pr(Z_{j-1} = r, Z_j = s | \mathbf{y}, \mathbf{x}; \boldsymbol{\Psi}^{(t-1)}) / \sum_{j=2}^M Pr(Z_{j-1} = r | \mathbf{y}, \mathbf{x}; \boldsymbol{\Psi}^{(t-1)})$$

- (b) Maximize Equation (5) with respect to $\boldsymbol{\delta}$ to obtain $\boldsymbol{\delta}^{(t)}$ such that $\sum_{l=1}^L \delta_l^{(t)} = 1$.
 - (c) Conditionally upon $\boldsymbol{\delta}^{(t)}$, maximize Equation (5) with respect to $\beta_1, \beta_3, \lambda_1, \lambda_3$ to obtain $\beta_1^{(t)}, \beta_3^{(t)}, \lambda_1^{(t)}, \lambda_3^{(t)}$,
 - (d) Iterate between (b) and (c) until convergence.
4. Iterate between 2. and 3. until convergence.

B4.1 Q -function of the EM algorithm

The Q -function of the EM algorithm is defined as $Q(\boldsymbol{\Psi} | \boldsymbol{\Psi}^{(t)}) = Q_0(\boldsymbol{\pi}, \boldsymbol{\gamma} | \boldsymbol{\Psi}^{(t)}) + Q_1(\boldsymbol{\psi}_1 | \boldsymbol{\Psi}^{(t)}) + Q_2(\boldsymbol{\delta}, \boldsymbol{\psi}_2 | \boldsymbol{\Psi}^{(t)}) + Q_3(\boldsymbol{\psi}_3 | \boldsymbol{\Psi}^{(t)})$, such that

$$\begin{aligned}
Q_0(\boldsymbol{\pi}, \boldsymbol{\gamma} | \boldsymbol{\Psi}^{(t)}) &= \sum_{r=1}^3 \left\{ Pr(Z_1 = r | \mathbf{y}, \mathbf{x}; \boldsymbol{\Psi}^{(t)}) \log(\pi_r) \right\} + \\
&\quad + \sum_{j=2}^M \sum_{r=1}^3 \sum_{s=1}^3 \left\{ Pr(Z_{j-1} = r, Z_j = s | \mathbf{y}, \mathbf{x}; \boldsymbol{\Psi}^{(t)}) \log(\gamma_{rs}) \right\}, \\
Q_1(\boldsymbol{\psi}_1 | \boldsymbol{\Psi}^{(t)}) &= \sum_{j=1}^M Pr(Z_j = 1 | \mathbf{y}, \mathbf{x}; \boldsymbol{\Psi}^{(t)}) \log f_1(\mathbf{y}_{..j} | \boldsymbol{\psi}_1), \\
Q_2(\boldsymbol{\delta}, \boldsymbol{\psi}_2 | \boldsymbol{\Psi}^{(t)}) &= \sum_{j=1}^M Pr(Z_j = 2 | \mathbf{y}, \mathbf{x}; \boldsymbol{\Psi}^{(t)}) \sum_{l=1}^L Pr(W_{jl} = 1 | Z_j = 2, \mathbf{y}_{..j}, \mathbf{x}; \boldsymbol{\Psi}^{(t)}) \times \\
&\quad \times \log \delta_l f_{(2,l)}(\mathbf{y}_{..j} | \mathbf{x}_l; \boldsymbol{\psi}_{(2,l)}), \quad \text{and} \\
Q_3(\boldsymbol{\psi}_3 | \boldsymbol{\Psi}^{(t)}) &= \sum_{j=1}^M Pr(Z_j = 3 | \mathbf{y}, \mathbf{x}; \boldsymbol{\Psi}^{(t)}) \log f_3(\mathbf{y}_{..j} | \boldsymbol{\psi}_3). \tag{B.4}
\end{aligned}$$

During the t^{th} iteration of the EM-algorithm, one solves $\arg \max_{\beta_1^{(t)}, \beta_3^{(t)}, \lambda_1^{(t)}, \lambda_3^{(t)}} Q_1(\boldsymbol{\psi}_1 | \boldsymbol{\Psi}^{(t)}) + Q_2(\boldsymbol{\delta}, \boldsymbol{\psi}_2 | \boldsymbol{\Psi}^{(t)}) + Q_3(\boldsymbol{\psi}_3 | \boldsymbol{\Psi}^{(t)})$ during the maximization step. This problem is analogous to estimating parameters in a weighted NB regression model, since Q_1 , Q_2 , and Q_3 can be seen as weighted log-likelihood functions from a NB generalized linear model with means and dispersions parametrized as described in the Section 3.1 of the main article.

The forward probabilities are defined as $f_{11}^p = \pi_1 f_1(\mathbf{y}_{..1} | \boldsymbol{\psi}_1)$, $f_{12}^p = \pi_2 f_2(\mathbf{y}_{..2} | \mathbf{x}; \boldsymbol{\delta}, \boldsymbol{\psi}_2)$, $f_{13}^p = \pi_3 f_3(\mathbf{y}_{..3} | \boldsymbol{\psi}_3)$, and, for $j = 1, \dots, M$, $f_{j1}^p = \sum_{l=1}^3 \gamma_{l1} f_{(j-1),l}^p f_1(\mathbf{y}_{..j} | \boldsymbol{\psi}_1)$, $f_{j2}^p = \sum_{l=1}^3 \gamma_{l2} f_{(j-1),l}^p f_2(\mathbf{y}_{..j} | \mathbf{x}; \boldsymbol{\delta}, \boldsymbol{\psi}_2)$, and $f_{j3}^p = \sum_{l=1}^3 \gamma_{l3} f_{(j-1),l}^p f_3(\mathbf{y}_{..j} | \boldsymbol{\psi}_3)$. Conversely, the backward probabilities are defined as $b_{Mk}^p = 1$, $\forall k = 1, 2, 3$, and, for $j = 1, \dots, M$, $b_{j1}^p = \sum_{l=1}^3 \gamma_{1l} b_{(j+1),l}^p f_1(\mathbf{y}_{..(j+1)} | \boldsymbol{\psi}_1)$, $b_{j2}^p = \sum_{l=1}^3 \gamma_{2l} b_{(j+1),l}^p f_2(\mathbf{y}_{..(j+1)} | \mathbf{x}; \boldsymbol{\delta}, \boldsymbol{\psi}_2)$, and $b_{j3}^p = \sum_{l=1}^3 \gamma_{3l} b_{(j+1),l}^p f_3(\mathbf{y}_{..(j+1)} | \boldsymbol{\psi}_3)$. Then, we have

$$\begin{aligned}
P(Z_j = k | \mathbf{y}, \mathbf{x}; \boldsymbol{\Psi}) &= \frac{f_{jk}^p b_{jk}^p}{\sum_{l=1}^3 f_{Ml}^p}, \quad \forall j = 1, \dots, M \quad \text{and} \quad k = 1, 2, 3, \\
P(Z_{j-1} = l, Z_j = 1 | \mathbf{y}, \mathbf{x}; \boldsymbol{\Psi}) &= \frac{f_{(j-1)l}^p \gamma_{l1} f_1(\mathbf{y}_{..j} | \boldsymbol{\psi}_1) b_{j1}^p}{\sum_{l=1}^3 f_{Ml}^p}, \\
P(Z_{j-1} = l, Z_j = 2 | \mathbf{y}, \mathbf{x}; \boldsymbol{\Psi}) &= \frac{f_{(j-1)l}^p \gamma_{l2} f_2(\mathbf{y}_{..j} | \mathbf{x}; \boldsymbol{\delta}, \boldsymbol{\psi}_2) b_{j2}^p}{\sum_{l=1}^3 f_{Ml}^p}, \\
P(Z_{j-1} = l, Z_j = 3 | \mathbf{y}, \mathbf{x}; \boldsymbol{\Psi}) &= \frac{f_{(j-1)l}^p \gamma_{l3} f_3(\mathbf{y}_{..j} | \boldsymbol{\psi}_3) b_{j3}^p}{\sum_{l=1}^3 f_{Ml}^p}, \quad \forall j = 2, \dots, M \text{ and } l = 1, 2, 3. \quad (\text{B.5})
\end{aligned}$$

Supporting Information C

In this appendix, we present an overview of the simulation scenarios and additional results.

C1 Read Count Simulation

In this section, we present additional results from the read count based simulation.

For each scenario, we simulated a hundred data sets as follows. First, we generated a sequence of hidden states with length M from a first-order Markov chain with 2^G states representing every combination of background and enrichment across G conditions. Here, we let states 1 and 2^G represent consensus background and consensus enrichment, respectively. Transition probabilities were chosen such that long sequences of enrichment regions (average length of 50 genomic windows) and equal proportions of differential states were generated, while the majority of the simulated states pertained to consensus background. Secondly, read counts were simulated for n replicates from G conditions following NB distributions with mean and dispersion parameters that led to a given SNR and respected the differential pattern of enrichment determined by the hidden state. For each scenario, this setup led to a simulated matrix of counts of dimension $M \times (n \times G)$. The aim of this study was to assess whether the presented model was able to aggregate all $2^G - 2$ simulated differential hidden states into its differential HMM state while maintaining a precise parameter estimation scheme. The model parameters used in this simulated study were estimated from ENCODE data pertaining to H3K36me3 and H3K27me3 histone modification marks and are shown in

the tables presented in this section.

Overall, while results were similar regarding different number of genome sizes and histone modification marks, the number of replicates played a major role in the performance of the model. In scenarios with either low SNR or high number of conditions, more replicates were needed to achieve better model performance. However, under the observed SNR, two replicates were sufficient to achieve a small relative bias regarding the estimation of most of the model parameters throughout all the evaluated scenarios. This is in agreement with the ENCODE Consortium guidelines, which recommends two technical or biological replicates to achieve best performance in ChIP-seq experiments.

Next, we assessed the sensitivity of our method to detect simulated differential regions of enrichment. First, differential regions were defined from the HMM posterior probabilities pertaining to the differential state by controlling the total FDR as defined in Section 3.2 (main text). For different nominal FDR threshold levels, the model sensitivity was estimated as the proportion of windows correctly assigned as differential out of the total number of simulated differential windows. Additionally, the observed FDR was calculated as the proportion of genomic windows incorrectly called as differential out of the total number of called differential windows. In Figures 15-16, panel A shows the average observed true positive rate (y-axis) and the observed FDR (x-axis) for different nominal FDR levels across a hundred simulated data. Results are shown for different levels of SNR, number of conditions, and number of replicates per condition. Overall, we observed that the number of replicates per condition played a major role on the sensitivity levels of the model, in which scenarios with two and four replicates had the best results regardless of the number of conditions and SNR levels. For scenarios with either high number of conditions or low SNR levels, more replicates were needed to achieve higher sensitivity.

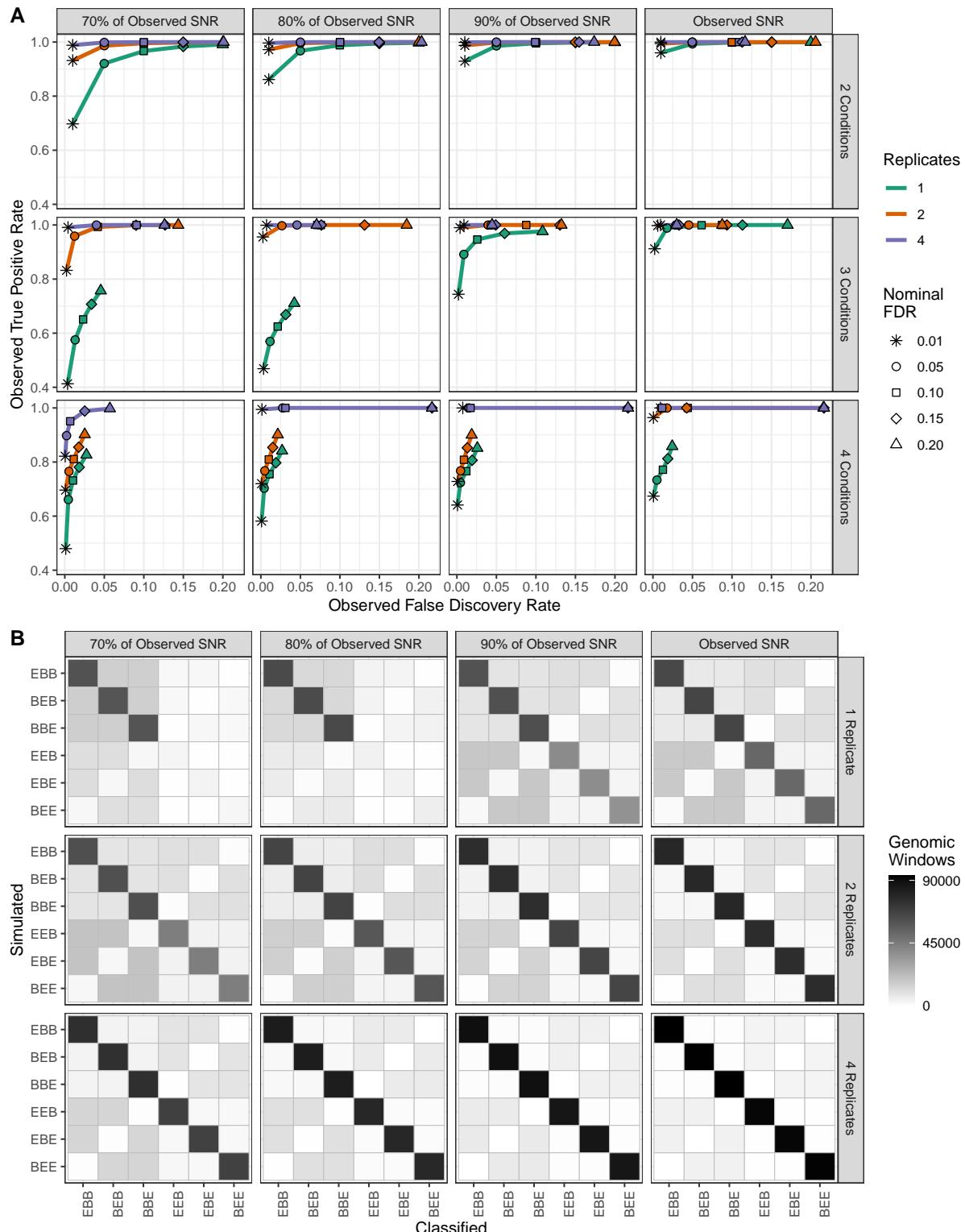
Web Table 2: Read count simulation. True values and average relative bias of parameter estimates (and 2.5^{th} , 97.5^{th} percentiles) across a hundred simulated data sets are shown for H3K27me3 with $1e+06$ genomic windows.

SNR	Conditions	Parameter	True	One Replicate		Two Replicates		Four Replicates	
				R. Bias	($P_{2.5}$, $P_{97.5}$)	R. Bias	($P_{2.5}$, $P_{97.5}$)	R. Bias	($P_{2.5}$, $P_{97.5}$)
Mean Ratio 2.24 (70% SNR)	Two	β_1	1.116	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)
		β_3	0.808	0.000	(-0.003, 0.003)	0.000	(-0.002, 0.002)	0.000	(-0.002, 0.001)
		λ_1	1.281	0.000	(-0.005, 0.005)	0.000	(-0.004, 0.002)	0.000	(-0.002, 0.002)
		λ_3	-0.232	-0.005	(-0.034, 0.029)	-0.002	(-0.030, 0.021)	-0.001	(-0.016, 0.013)
		β_1	1.116	-0.002	(-0.003, -0.000)	-0.003	(-0.004, -0.002)	-0.001	(-0.001, 0.000)
	Three	β_3	0.808	-0.122	(-0.129, -0.114)	-0.012	(-0.015, -0.009)	-0.001	(-0.002, 0.000)
		λ_1	1.281	0.017	(0.011, 0.022)	0.004	(0.001, 0.007)	0.000	(-0.002, 0.002)
		λ_3	-0.232	0.857	(0.805, 0.913)	0.113	(0.089, 0.142)	0.010	(-0.005, 0.023)
		β_1	1.116	0.000	(-0.002, 0.001)	-0.012	(-0.013, -0.011)	-0.011	(-0.012, -0.009)
		β_3	0.808	-0.135	(-0.141, -0.128)	-0.088	(-0.091, -0.085)	-0.032	(-0.054, -0.023)
Mean Ratio 2.57(80% SNR)	Four	λ_1	1.281	0.018	(0.012, 0.024)	0.028	(0.025, 0.031)	0.017	(0.014, 0.023)
		λ_3	-0.232	0.924	(0.869, 0.977)	0.770	(0.744, 0.798)	0.367	(0.276, 0.542)
		β_1	1.116	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)
		β_3	0.942	0.000	(-0.002, 0.002)	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)
		λ_1	1.281	0.000	(-0.004, 0.004)	0.000	(-0.003, 0.003)	0.000	(-0.002, 0.002)
	Three	λ_3	-0.099	0.001	(-0.069, 0.089)	-0.002	(-0.052, 0.056)	0.001	(-0.034, 0.045)
		β_1	1.116	-0.002	(-0.004, -0.000)	-0.002	(-0.003, -0.001)	0.000	(-0.001, 0.000)
		β_3	0.942	-0.133	(-0.142, -0.122)	-0.003	(-0.004, -0.001)	0.000	(-0.001, 0.001)
		λ_1	1.281	0.012	(0.003, 0.018)	0.001	(-0.002, 0.005)	0.000	(-0.002, 0.002)
		λ_3	-0.099	3.091	(2.870, 3.267)	0.120	(0.066, 0.182)	0.009	(-0.028, 0.041)
Mean Ratio 2.89(90% SNR)	Four	β_1	1.116	-0.005	(-0.006, -0.003)	-0.016	(-0.017, -0.015)	-0.002	(-0.002, -0.001)
		β_3	0.942	-0.125	(-0.131, -0.122)	-0.082	(-0.085, -0.078)	-0.001	(-0.002, -0.000)
		λ_1	1.281	0.009	(0.004, 0.013)	0.027	(0.024, 0.031)	0.002	(-0.000, 0.004)
		λ_3	-0.099	2.934	(2.834, 3.048)	2.515	(2.432, 2.591)	0.077	(0.037, 0.116)
		β_1	1.116	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)
	Three	β_3	1.060	0.000	(-0.002, 0.002)	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)
		λ_1	1.281	0.000	(-0.005, 0.004)	0.000	(-0.003, 0.003)	0.000	(-0.002, 0.002)
		λ_3	0.019	0.024	(-0.338, 0.388)	-0.014	(-0.263, 0.246)	-0.008	(-0.197, 0.163)
		β_1	1.116	-0.005	(-0.007, -0.002)	-0.001	(-0.002, 0.000)	0.000	(-0.001, 0.001)
		β_3	1.060	-0.035	(-0.126, -0.019)	-0.001	(-0.002, 0.000)	0.000	(-0.001, 0.001)
Mean Ratio 2.89(90% SNR)	Four	λ_1	1.281	-0.001	(-0.006, 0.005)	0.001	(-0.002, 0.004)	0.000	(-0.002, 0.003)
		λ_3	0.019	-6.431	(-20.113, -3.826)	-0.285	(-0.561, -0.062)	-0.016	(-0.202, 0.155)
		β_1	1.116	-0.010	(-0.012, -0.008)	-0.019	(-0.020, -0.018)	0.000	(-0.001, 0.000)
		β_3	1.060	-0.117	(-0.121, -0.113)	-0.075	(-0.077, -0.072)	0.000	(-0.001, 0.001)
		λ_1	1.281	-0.001	(-0.007, 0.004)	0.026	(0.024, 0.030)	0.000	(-0.002, 0.002)
	Three	λ_3	0.019	-19.074	(-19.650, -18.359)	-16.439	(-16.929, -15.892)	-0.052	(-0.181, 0.094)
		β_1	1.116	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)
		β_3	1.165	0.000	(-0.002, 0.001)	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)
		λ_1	1.281	0.000	(-0.004, 0.004)	0.000	(-0.003, 0.003)	0.000	(-0.002, 0.002)
		λ_3	0.124	0.004	(-0.057, 0.066)	-0.004	(-0.047, 0.038)	-0.003	(-0.035, 0.034)
Mean Ratio 3.21(100% SNR)	Four	β_1	1.116	-0.003	(-0.005, -0.002)	0.000	(-0.001, 0.000)	0.000	(-0.001, 0.001)
		β_3	1.165	-0.007	(-0.010, -0.005)	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)
		λ_1	1.281	-0.001	(-0.006, 0.003)	0.001	(-0.002, 0.004)	0.000	(-0.002, 0.002)
		λ_3	0.124	-0.317	(-0.404, -0.223)	-0.023	(-0.057, 0.012)	-0.003	(-0.025, 0.024)
		β_1	1.116	-0.014	(-0.015, -0.012)	-0.007	(-0.008, -0.006)	0.000	(-0.001, 0.000)
	Three	β_3	1.165	-0.111	(-0.114, -0.109)	-0.003	(-0.004, -0.002)	0.000	(-0.001, 0.001)
		λ_1	1.281	-0.012	(-0.016, -0.007)	0.007	(0.004, 0.010)	0.000	(-0.001, 0.002)
		λ_3	0.124	-3.520	(-3.589, -3.439)	-0.360	(-0.446, -0.282)	-0.005	(-0.025, 0.017)

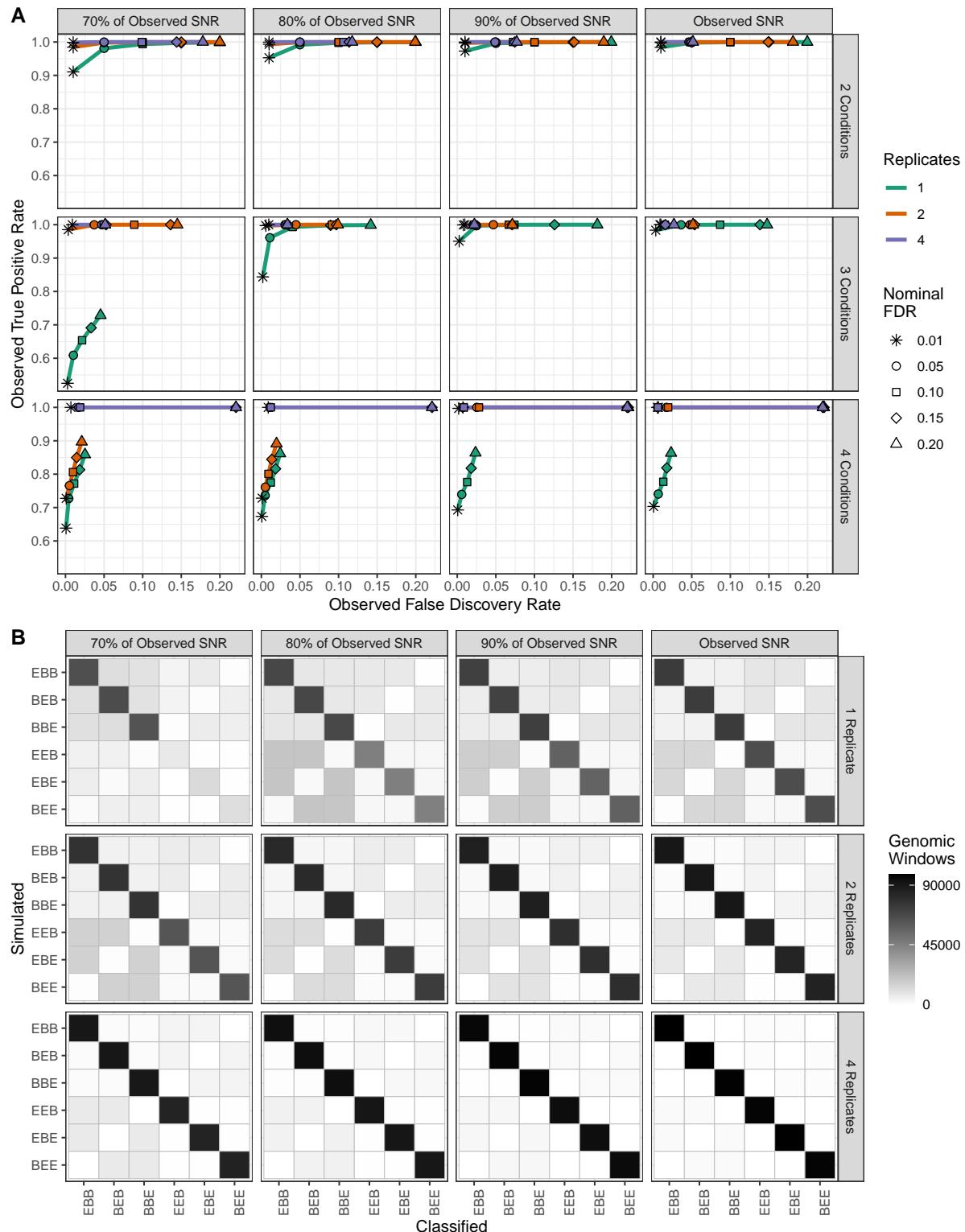
Web Table 3: Read count simulation. True values and average relative bias of parameter estimates (and 2.5th, 97.5th percentiles) across a hundred simulated data sets are shown for H3K36me3 with 1e+06 genomic windows.

SNR	Conditions	Parameter	True	One Replicate		Two Replicates		Four Replicates	
				R. Bias	(P _{2.5} , P _{97.5})	R. Bias	(P _{2.5} , P _{97.5})	R. Bias	(P _{2.5} , P _{97.5})
Mean Ratio 3.68 (70% SNR)	Two	β_1	0.756	0.000	(-0.002, 0.002)	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)
		β_3	1.304	0.000	(-0.002, 0.002)	0.000	(-0.001, 0.002)	0.000	(-0.001, 0.001)
		λ_1	1.173	0.000	(-0.006, 0.006)	0.000	(-0.004, 0.003)	-0.001	(-0.003, 0.002)
	Three	λ_3	-0.467	0.000	(-0.015, 0.020)	0.000	(-0.012, 0.012)	-0.001	(-0.010, 0.010)
		β_1	0.756	0.003	(0.001, 0.005)	-0.002	(-0.003, -0.000)	0.000	(-0.001, 0.001)
		β_3	1.304	-0.115	(-0.122, -0.106)	-0.001	(-0.002, -0.000)	0.000	(-0.001, 0.000)
Mean Ratio 4.21(80% SNR)	Two	λ_1	1.173	0.028	(0.020, 0.036)	0.002	(-0.001, 0.006)	0.000	(-0.003, 0.002)
		λ_3	-0.467	0.693	(0.651, 0.730)	0.017	(0.006, 0.030)	0.000	(-0.007, 0.008)
		β_1	0.756	-0.009	(-0.012, -0.007)	-0.020	(-0.021, -0.018)	-0.001	(-0.002, 0.000)
	Three	β_3	1.304	-0.107	(-0.109, -0.103)	-0.073	(-0.075, -0.070)	0.000	(-0.001, 0.001)
		λ_1	1.173	0.052	(0.045, 0.058)	0.046	(0.042, 0.050)	0.001	(-0.002, 0.003)
		λ_3	-0.467	0.736	(0.715, 0.757)	0.573	(0.558, 0.589)	0.004	(-0.004, 0.011)
Mean Ratio 4.74(90% SNR)	Two	β_1	0.756	0.000	(-0.002, 0.002)	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)
		β_3	1.438	0.000	(-0.002, 0.002)	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)
		λ_1	1.173	0.000	(-0.005, 0.005)	0.000	(-0.004, 0.004)	0.000	(-0.003, 0.002)
	Three	λ_3	-0.334	0.000	(-0.019, 0.024)	0.001	(-0.015, 0.017)	-0.001	(-0.012, 0.010)
		β_1	0.756	-0.008	(-0.010, -0.006)	-0.001	(-0.002, 0.000)	0.000	(-0.001, 0.001)
		β_3	1.438	-0.014	(-0.019, -0.010)	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)
Mean Ratio 5.26(100% SNR)	Two	λ_1	1.173	0.007	(0.002, 0.013)	0.001	(-0.002, 0.004)	0.000	(-0.002, 0.002)
		λ_3	-0.334	0.229	(0.167, 0.279)	0.010	(-0.003, 0.025)	0.001	(-0.008, 0.008)
		β_1	0.756	-0.016	(-0.018, -0.013)	-0.023	(-0.025, -0.022)	0.000	(-0.001, 0.001)
	Three	β_3	1.438	-0.101	(-0.103, -0.099)	-0.067	(-0.069, -0.065)	0.000	(-0.001, 0.001)
		λ_1	1.173	0.037	(0.032, 0.045)	0.042	(0.036, 0.046)	0.000	(-0.002, 0.002)
		λ_3	-0.334	1.237	(1.207, 1.273)	0.978	(0.949, 1.003)	0.001	(-0.008, 0.008)
Mean Ratio 5.26(100% SNR)	Two	β_1	0.756	0.000	(-0.003, 0.002)	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)
		β_3	1.556	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)
		λ_1	1.173	0.000	(-0.007, 0.005)	0.000	(-0.004, 0.005)	0.000	(-0.003, 0.003)
	Three	λ_3	-0.216	-0.002	(-0.041, 0.032)	0.001	(-0.022, 0.032)	0.001	(-0.016, 0.020)
		β_1	0.756	-0.005	(-0.007, -0.003)	0.000	(-0.002, 0.001)	0.000	(-0.001, 0.001)
		β_3	1.556	-0.004	(-0.005, -0.002)	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)
Mean Ratio 5.26(100% SNR)	Two	λ_1	1.173	0.004	(-0.002, 0.010)	0.000	(-0.004, 0.004)	0.000	(-0.002, 0.002)
		λ_3	-0.216	0.158	(0.112, 0.197)	0.008	(-0.015, 0.031)	0.001	(-0.013, 0.014)
		β_1	0.756	-0.023	(-0.026, -0.021)	-0.004	(-0.006, -0.002)	0.000	(-0.001, 0.001)
	Three	β_3	1.556	-0.095	(-0.098, -0.093)	-0.001	(-0.001, 0.000)	0.000	(-0.001, 0.000)
		λ_1	1.173	0.024	(0.016, 0.030)	0.005	(0.001, 0.008)	0.000	(-0.002, 0.002)
		λ_3	-0.216	2.219	(2.167, 2.278)	0.072	(0.050, 0.098)	0.002	(-0.011, 0.014)
Mean Ratio 5.26(100% SNR)	Two	β_1	0.756	0.000	(-0.002, 0.002)	0.000	(-0.002, 0.002)	0.000	(-0.001, 0.001)
		β_3	1.661	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.001)
		λ_1	1.173	0.000	(-0.005, 0.006)	0.000	(-0.004, 0.004)	0.000	(-0.002, 0.003)
	Three	λ_3	-0.111	0.002	(-0.063, 0.085)	0.004	(-0.046, 0.055)	-0.001	(-0.034, 0.031)
		β_1	0.756	-0.003	(-0.005, -0.001)	0.000	(-0.002, 0.001)	0.000	(-0.001, 0.001)
		β_3	1.661	-0.001	(-0.003, -0.000)	0.000	(-0.001, 0.001)	0.000	(-0.000, 0.001)
Mean Ratio 5.26(100% SNR)	Two	λ_1	1.173	0.002	(-0.003, 0.007)	0.000	(-0.004, 0.004)	0.000	(-0.002, 0.003)
		λ_3	-0.111	0.164	(0.084, 0.236)	0.008	(-0.044, 0.048)	0.004	(-0.022, 0.036)
		β_1	0.756	-0.030	(-0.032, -0.027)	-0.001	(-0.003, 0.000)	0.000	(-0.001, 0.001)
	Three	β_3	1.661	-0.089	(-0.092, -0.087)	0.000	(-0.001, 0.001)	0.000	(-0.001, 0.000)
		λ_1	1.173	0.014	(0.008, 0.020)	0.002	(-0.002, 0.006)	0.000	(-0.002, 0.002)
		λ_3	-0.111	4.934	(4.824, 5.048)	0.044	(0.007, 0.093)	0.003	(-0.021, 0.030)

Finally, we used the estimated mixture model posterior probabilities $Pr(W_{jl} = 1|Z_j = 2, \mathbf{y}_{..j}, \mathbf{x}; \hat{\boldsymbol{\Psi}})$, $j = 1, \dots, M$, to classify the differential combinatorial patterns of enrichment of detected differential windows. To this end, we first calculated the maximum estimated mixture model posterior probability across all L components to determine the most likely differential combinatorial pattern from genomic windows assigned to be part of the differential state. Then, we compared the window-based classification with the true window-based simulated states from the Markov Chain (states $2, \dots, G - 1$). In Figures 15-16, panel B shows the confusion matrices of classified (x-axis) and simulated (y-axis) differential windows for a scenario with three conditions. Differential combinatorial patterns of enrichment are represented by the sequences of letters 'E' (enrichment) and 'B' (background), such that each letter corresponds to the status of a given condition. The number of windows (averaged over all simulated data sets) is shown as entries of the matrices and represented by the color scale. Darker colors on the diagonal entries indicate better agreement between simulated and classified patterns. By utilizing the posterior probabilities from the mixture model, we observed a good performance when classifying the differential combinatorial pattern of enrichment from differential windows. Results were best under scenarios with higher number of replicates or SNR.



Web Figure 15: Read count simulation. (A): average observed TPR and FDR for different nominal FDR threshold levels for various scenarios in simulated data of H3K27me3 with 10^6 genomic windows. (B): confusion matrices are shown for the 3 conditions scenario. On x- and y-axes the labels indicate the differential classified and simulated patterns, respectively. For instance, EBE denotes enrichment in conditions 1 and 3 only. Darker colors on the diagonal of the matrix indicate better agreement.



Web Figure 16: Read count simulation. (A): average observed TPR and FDR for different nominal FDR threshold levels for various scenarios in simulated data of H3K36me3 with 10^5 genomic windows. (B): confusion matrices are shown for the 3 conditions scenario. On x- and y-axes the labels indicate the differential classified and simulated patterns, respectively. For instance, EBE denotes enrichment in conditions 1 and 3 only. Darker colors on the diagonal of the matrix indicate better agreement.

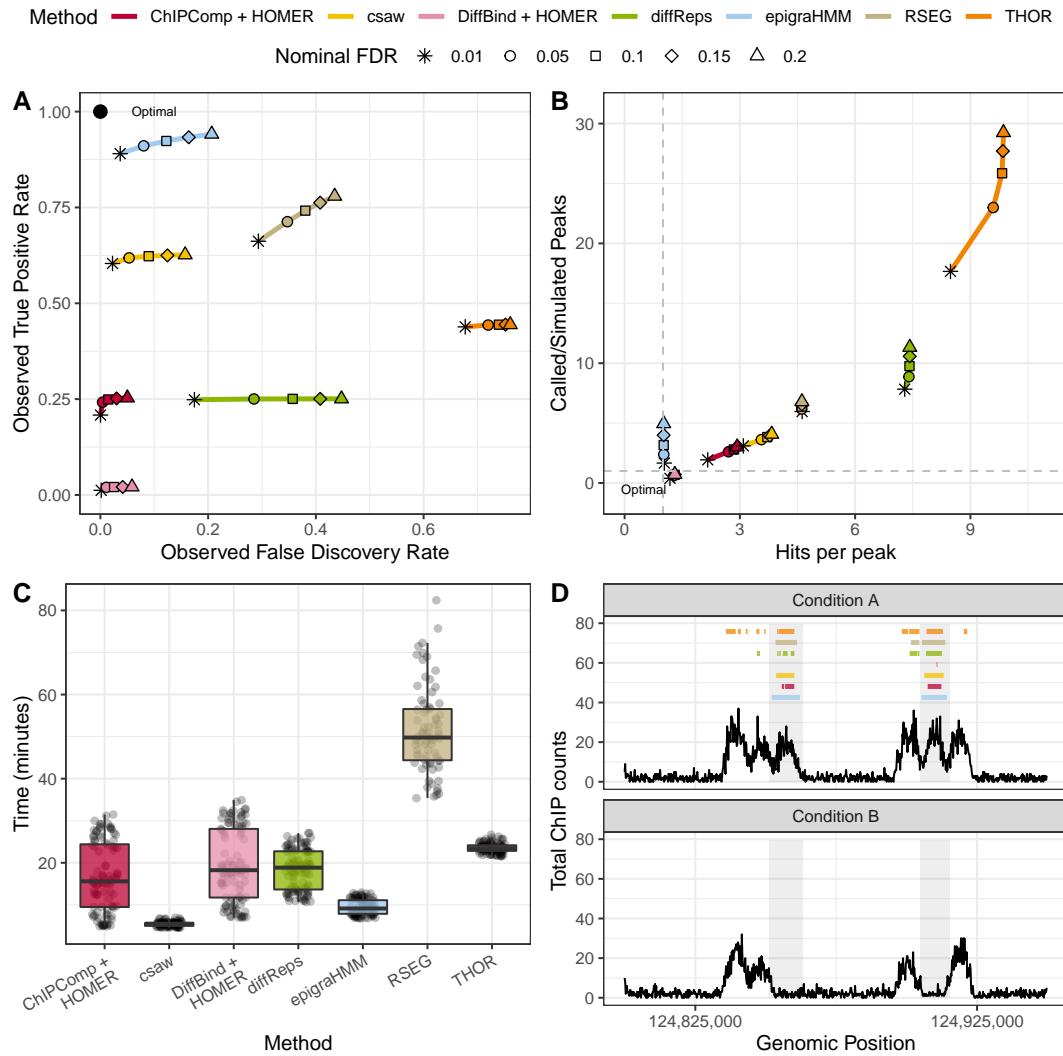
C2 Sequencing Read Simulation

In this section, we present additional results from the sequencing read simulation using different FDR control threshold for differential peak call.

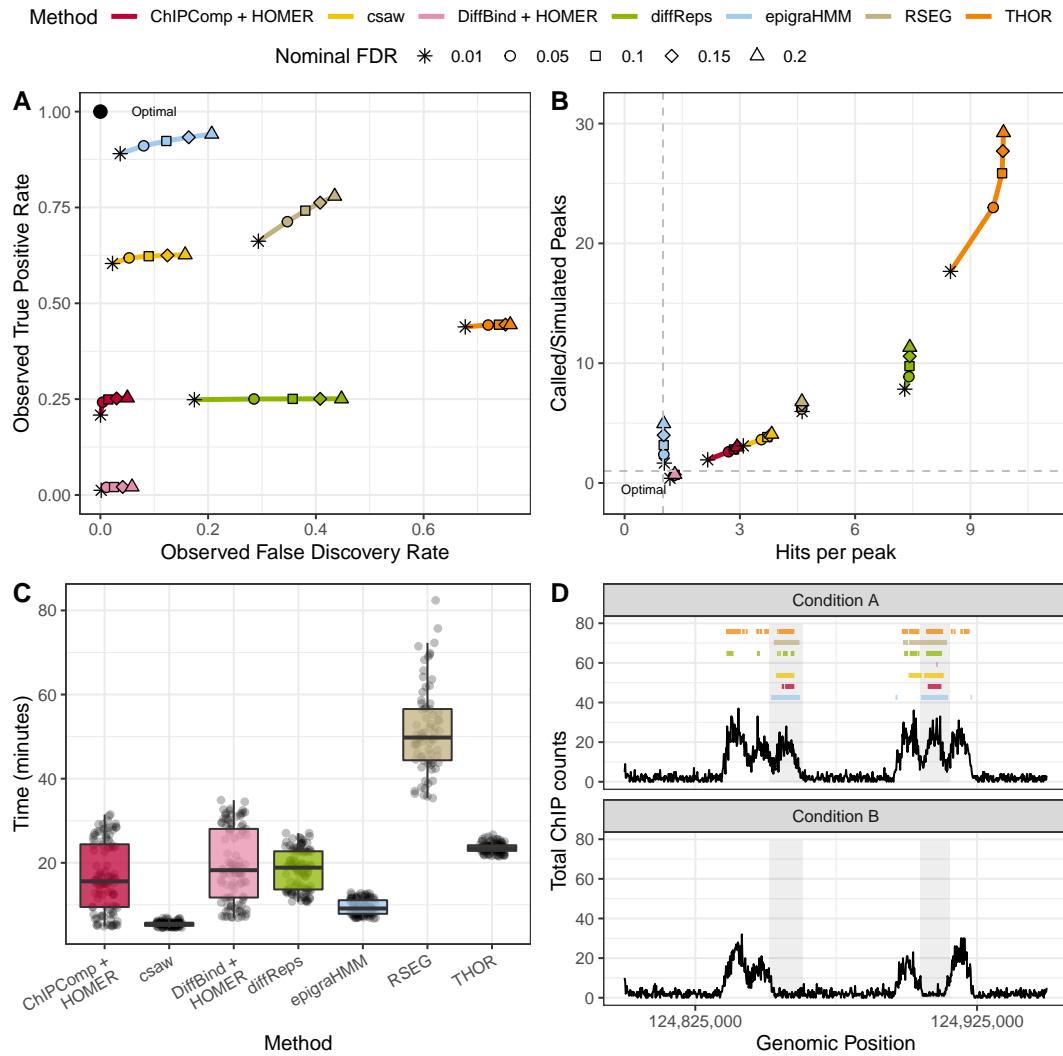
We performed a second simulation study aiming to compare the proposed model with the current DPCs ChIPComp, csaw, DiffBind, diffReps, RSEG, and THOR, since none of them was designed to accept matrices of window-based read counts as input. To this purpose, we revisited the simulation study presented by [7] where data were generated in a more general scheme without a particular read count model assumption. Here, sequencing reads from ChIP-seq experiments were generated for two conditions and two replicates per condition. For the differential peaks callers ChIPComp and DiffBind that require sets of candidate regions, we followed the analyses presented by [7] and called peaks in advance using HOMER. Those were then used as input in the respective software for differential call. For ChIPComp and DiffBind, we attempted to call peaks in advance using MACS2. However, MACS2. exhibited issues to work on simulated data as it was not able to detect peaks. Here, we adapted their simulation pipeline aiming to generate broad and diffuse differential regions of enrichment. Specifically, a total of 4000 binding sites spaced 60-65 kbp apart were simulated such that each site was formed by three neighboring peaks 10kb wide. For each peak, the number of sequencing reads were simulated from a NB distribution with mean 300 and dispersions sampled from an inverse chi-squared distribution with 20 degrees of freedom. Reads were positioned on peak sites to form smooth binding profiles. To introduce differential enrichment, one or two peaks from 1000 binding sites were randomly chosen and reads pertaining to them were removed from the sequencing library in one of the two conditions. This process was repeated for the second condition, which led to a total of 2000 differential peaks. Background reads were added to the library to maintain the realistic aspects of the experiment. A hundred simulated data sets were generated and peaks were called by all the methods under multiple nominal FDR thresholds. For our method and RSEG, window-based posterior probabilities were used to control the total FDR. For

illustration purposes, we refer to the method presented in this article as epigraHMM.

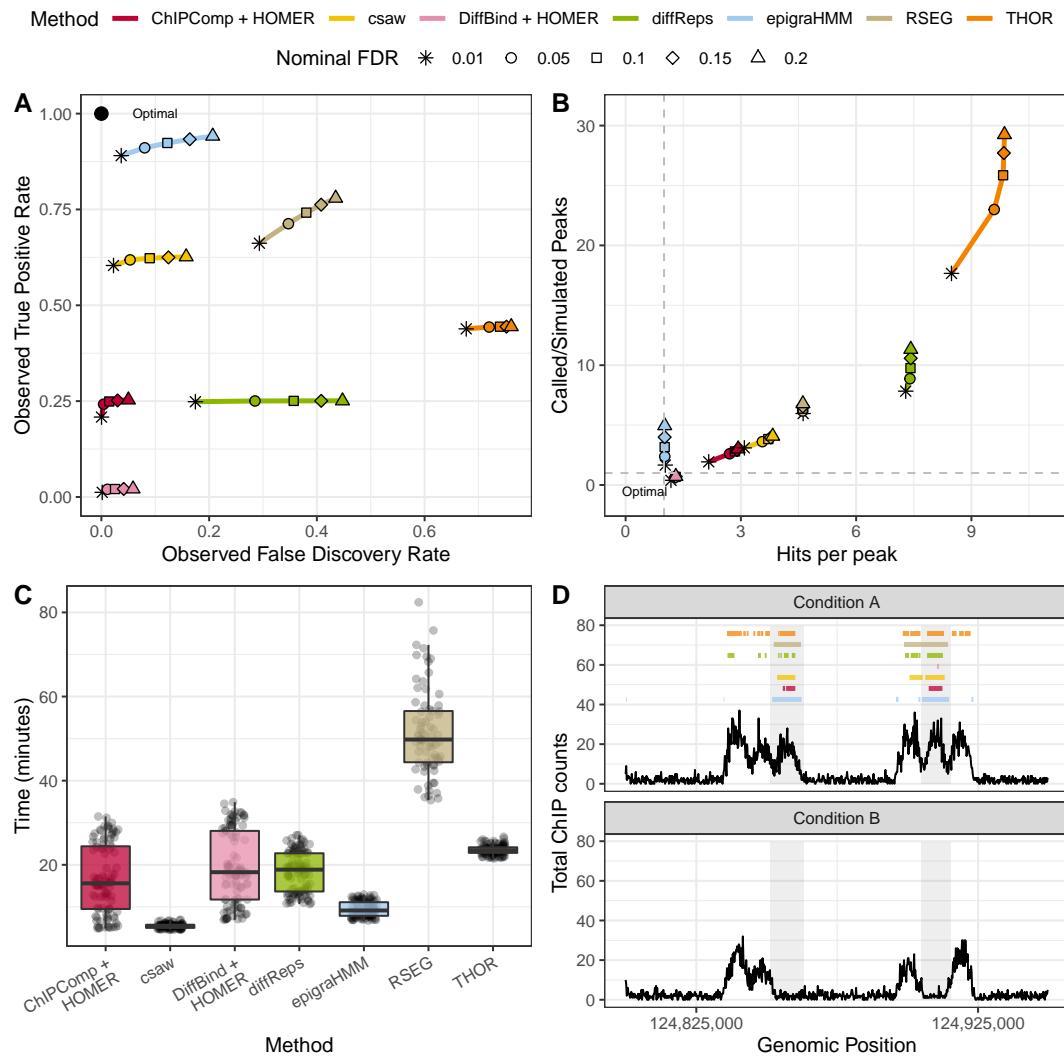
Specifically, panel D from Figures 17, 18, 19, and 20 show example of peak calls using FDR control cutoffs 0.01, 0.10, 0.15, and 0.20, respectively. Overall, results were consistent across different choices of FDR control. This fact is also shown by the observed sensitivity and false discovery rate from all methods (panels A). We observed a clear distinction of methods regarding the observed sensitivity and specificity such that RSEG, diffReps, and THOR exhibited the highest observed FDR and ChIPComp and DiffBind exhibited the lowest sensitivity.



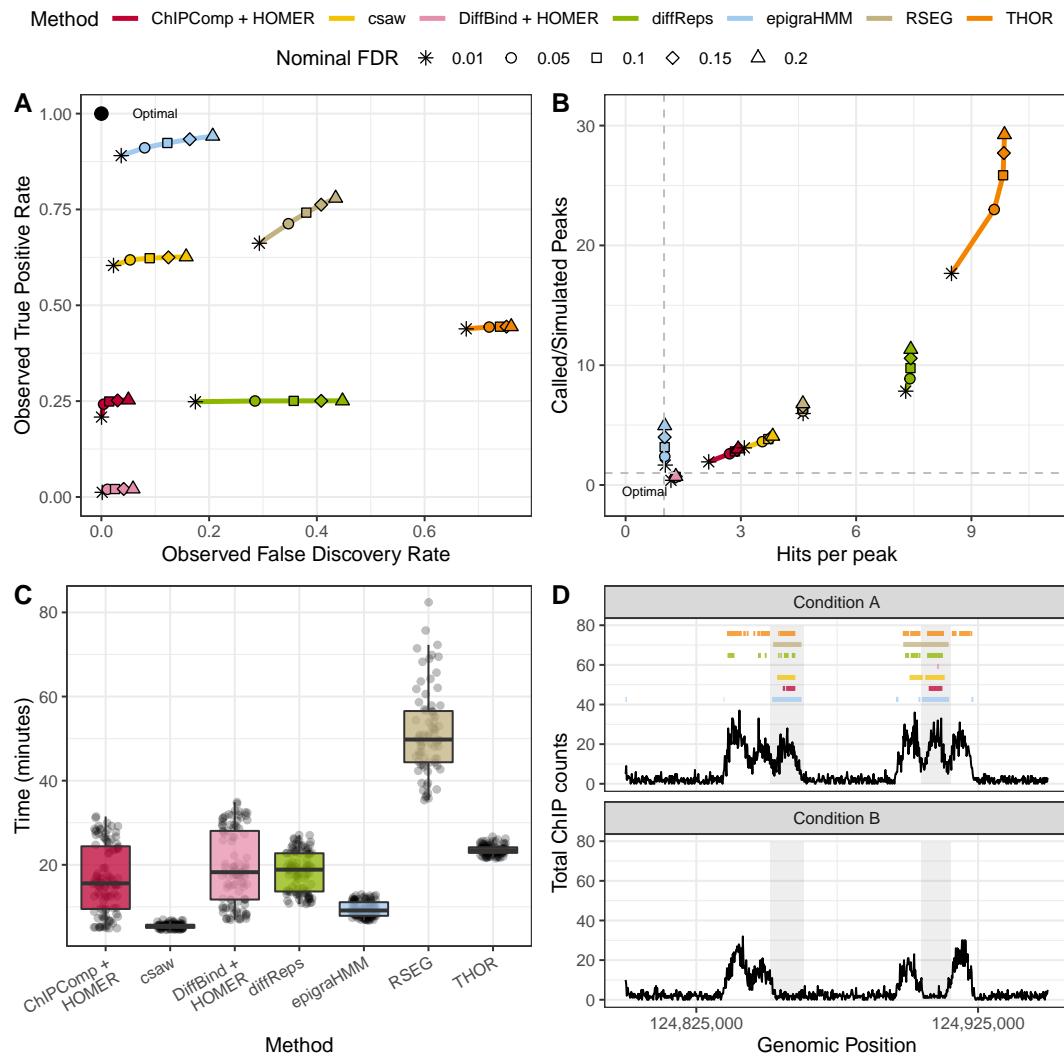
Web Figure 17: Sequencing read simulation. (A): average observed sensitivity and FDR for various methods under different nominal FDR levels. (B): scatter plot of average ratio of called and simulated peaks (y-axis) and average number of called peaks intersecting true differential regions (x-axis) for different nominal FDR levels. (C): box plot of computing time (in minutes) for various algorithms. (D): an example of differential peak calls under a nominal FDR threshold 0.01. Shaded areas indicate true differential peaks.



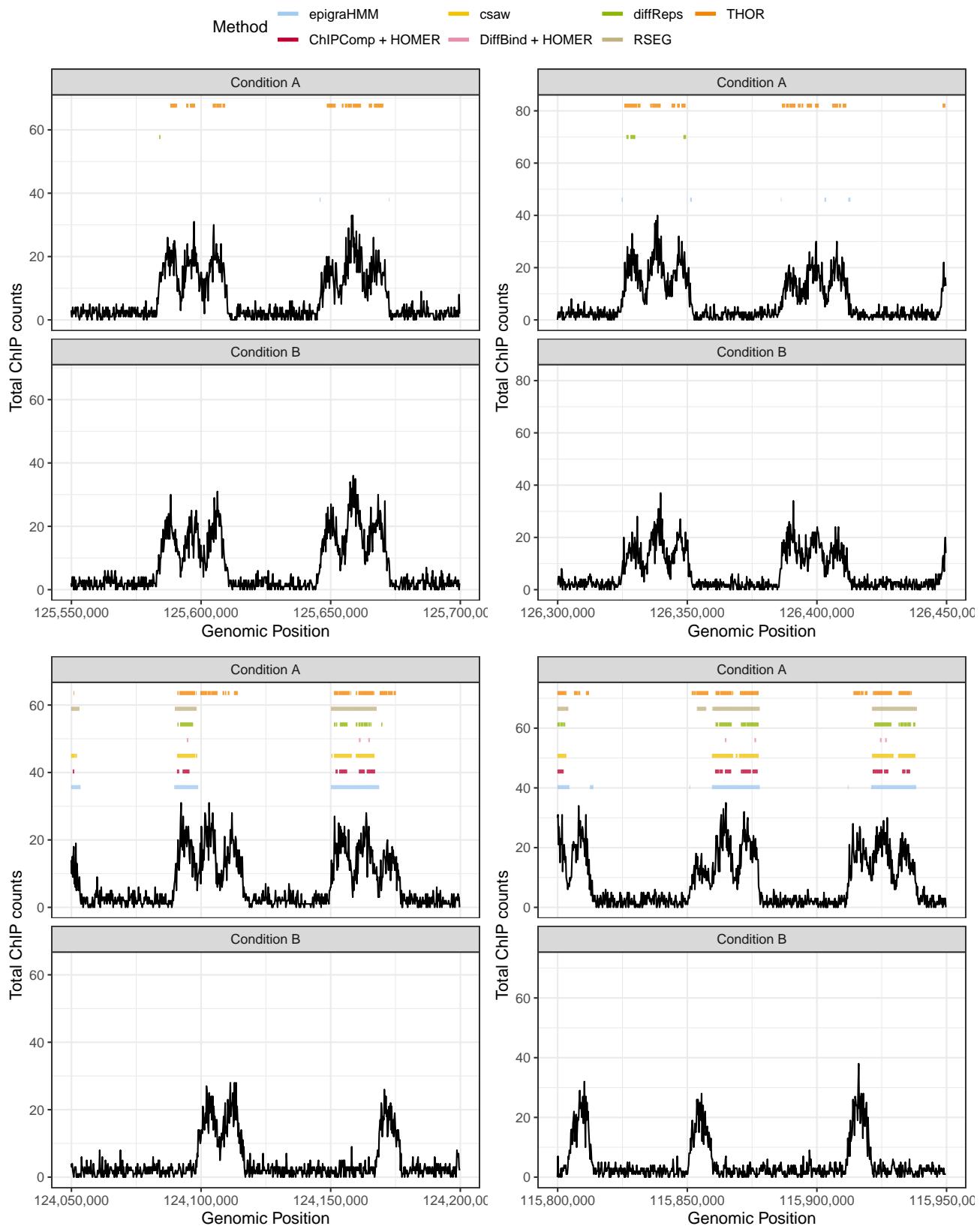
Web Figure 18: Sequencing read simulation. (A): average observed sensitivity and FDR for various methods under different nominal FDR levels. (B): scatter plot of average ratio of called and simulated peaks (y-axis) and average number of called peaks intersecting true differential regions (x-axis) for different nominal FDR levels. (C): box plot of computing time (in minutes) for various algorithms. (D): an example of differential peak calls under a nominal FDR threshold 0.10. Shaded areas indicate true differential peaks.



Web Figure 19: Sequencing read simulation. (A): average observed sensitivity and FDR for various methods under different nominal FDR levels. (B): scatter plot of average ratio of called and simulated peaks (y-axis) and average number of called peaks intersecting true differential regions (x-axis) for different nominal FDR levels. (C): box plot of computing time (in minutes) for various algorithms. (D): an example of differential peak calls under a nominal FDR threshold 0.15. Shaded areas indicate true differential peaks.



Web Figure 20: Sequencing read simulation. (A): average observed sensitivity and FDR for various methods under different nominal FDR levels. (B): scatter plot of average ratio of called and simulated peaks (y-axis) and average number of called peaks intersecting true differential regions (x-axis) for different nominal FDR levels. (C): box plot of computing time (in minutes) for various algorithms. (D): an example of differential peak calls under a nominal FDR threshold 0.20. Shaded areas indicate true differential peaks.



Web Figure 21: Sequencing read simulation. Examples of differential called peaks from the various methods under an FDR control of 0.05.

Supporting Information D

In this appendix, we present an overview of the application on data from the ENCODE Consortium.

D1 RNA-seq data and gene transcription

Under the three cell lines scenario (presented in the main text), we used RNA-seq experimental data from the ENCODE Consortium on Helas3, Hepg2, and Huvec human cells to quantify the expression levels of genes as follows. First, we used Salmon [9] to quantify transcript expression from cell-specific RNA-seq experiments. We then calculated, using the R package *tximport* [12], estimated counts using abundance estimates (transcripts per million, TPM) scaled up to the average transcript length over samples and library size. This step ensures that counts computed from Salmon are not correlated with the average transcript length. Secondly, we calculated the number of ChIP-seq reads from H3K4me3 and H3K27ac overlapping gene promoters. Promoter regions extend around the transcription start site 2000 base pairs upstream and 200 base pair downstream. Finally, read counts from RNA-seq and ChIP-seq were normalized for sequencing depth using DESeq2.

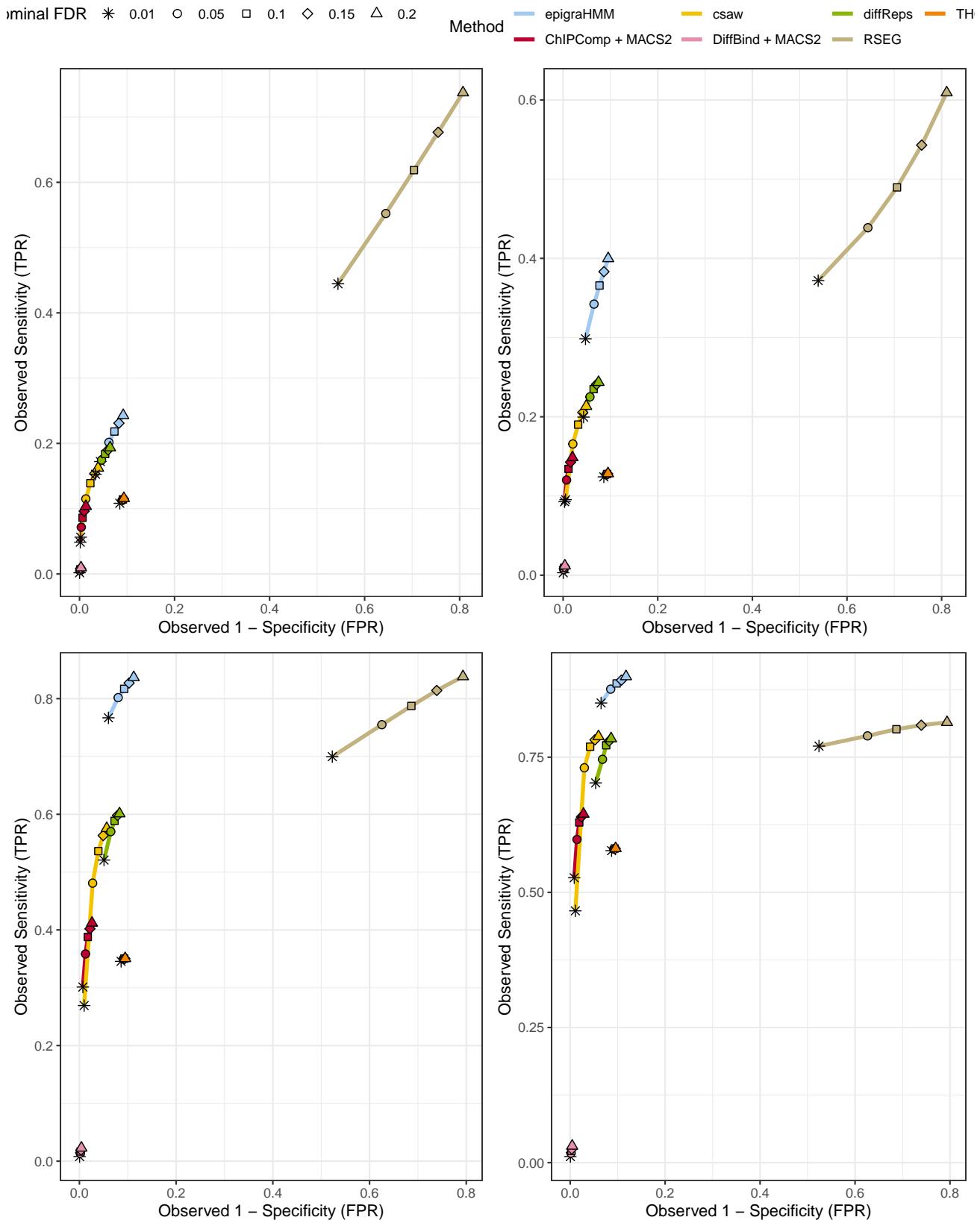
For the two cell line scenario, differentially transcribed genes were defined through log2 fold changes of H3K36me3 ChIP-seq read counts. We observed several cases in which differentially expressed genes (defined by RNA-seq data) did not exhibit differential enrichment for H3K36me3. However, due to the activating roles of H3K36me3 on gene transcription, we

follow the ideas presented by [14] and [4] and defined differentially transcribed genes based on log2 fold changes of H3K36me3 ChIP-seq read counts.

Sensitivity and specificity metrics were calculated on the window-level as follows. For non-overlapping genomic windows b_1, \dots, b_M , let $g_j = I(b_j \in \text{differentially transcribed gene})$ and $d_j = I(b_j \in \text{differential peak})$ denote the indicators of genomic windows being associated with either differential gene bodies or differential peaks, respectively, for $j = 1, \dots, M$, for a given method and nominal FDR level. Then, the observed sensitivity (TPR), specificity (1-FPR), and FDR were calculated as follows:

$$\begin{aligned}\text{Sensitivity} &= \frac{\sum_{j=1}^M g_j d_j}{\sum_{j=1}^M g_j} \\ \text{Specificity} &= \frac{\sum_{j=1}^M (1 - g_j)(1 - d_j)}{\sum_{j=1}^M (1 - g_j)} \\ \text{FDR} &= \frac{\sum_{j=1}^M (1 - g_j)d_j}{\sum_{j=1}^M d_j}\end{aligned}$$

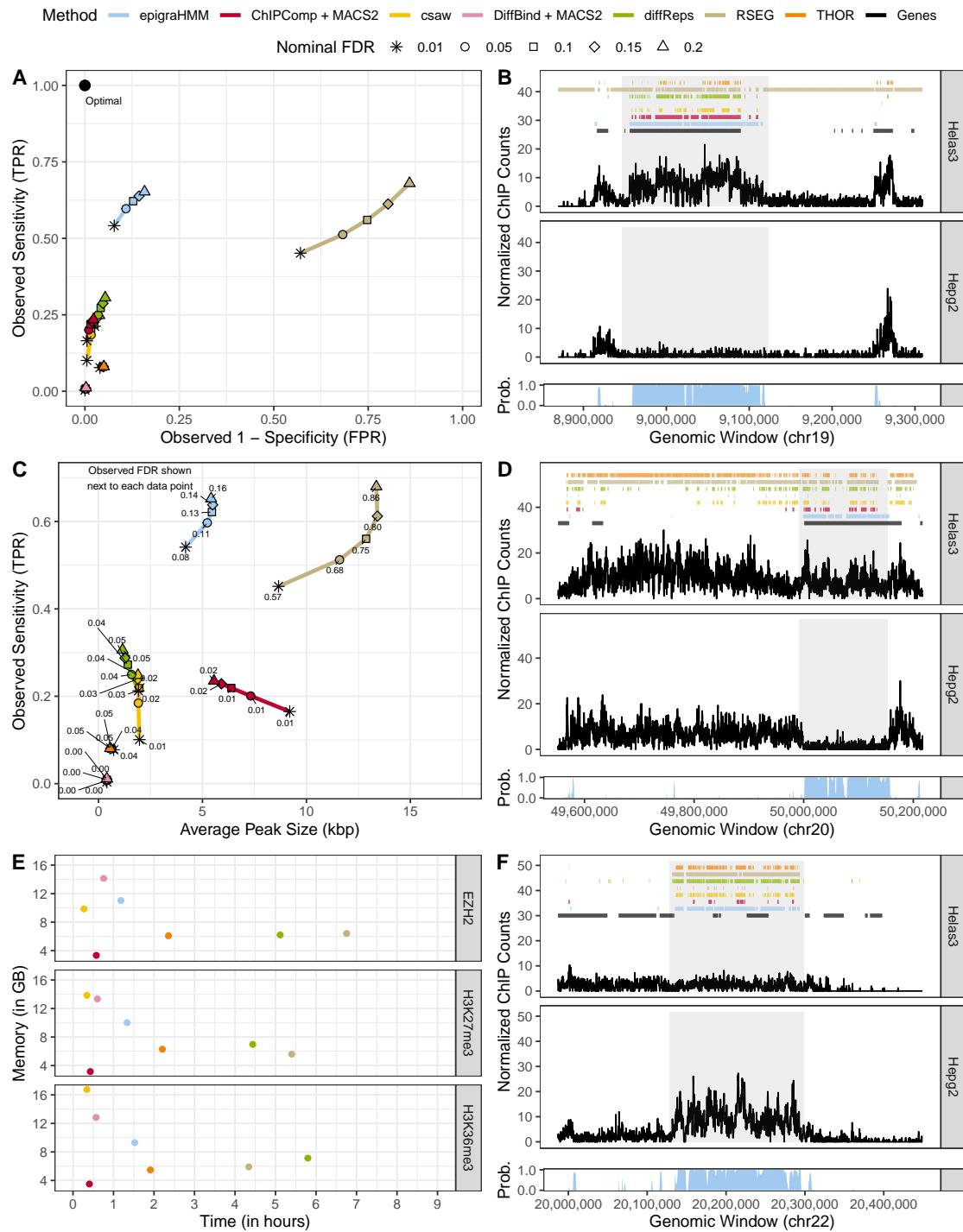
We present additional results using log2 FC 1, 1.5, 2.5, and 3 in Figure 22. Overall the presented model outperformed all current methods regardless of the choice of FC cutoff, as we maintained the highest sensitivity for a limited observed false discovery rate. RSEG showed significant higher observed false discovery rate in all the scenarios.



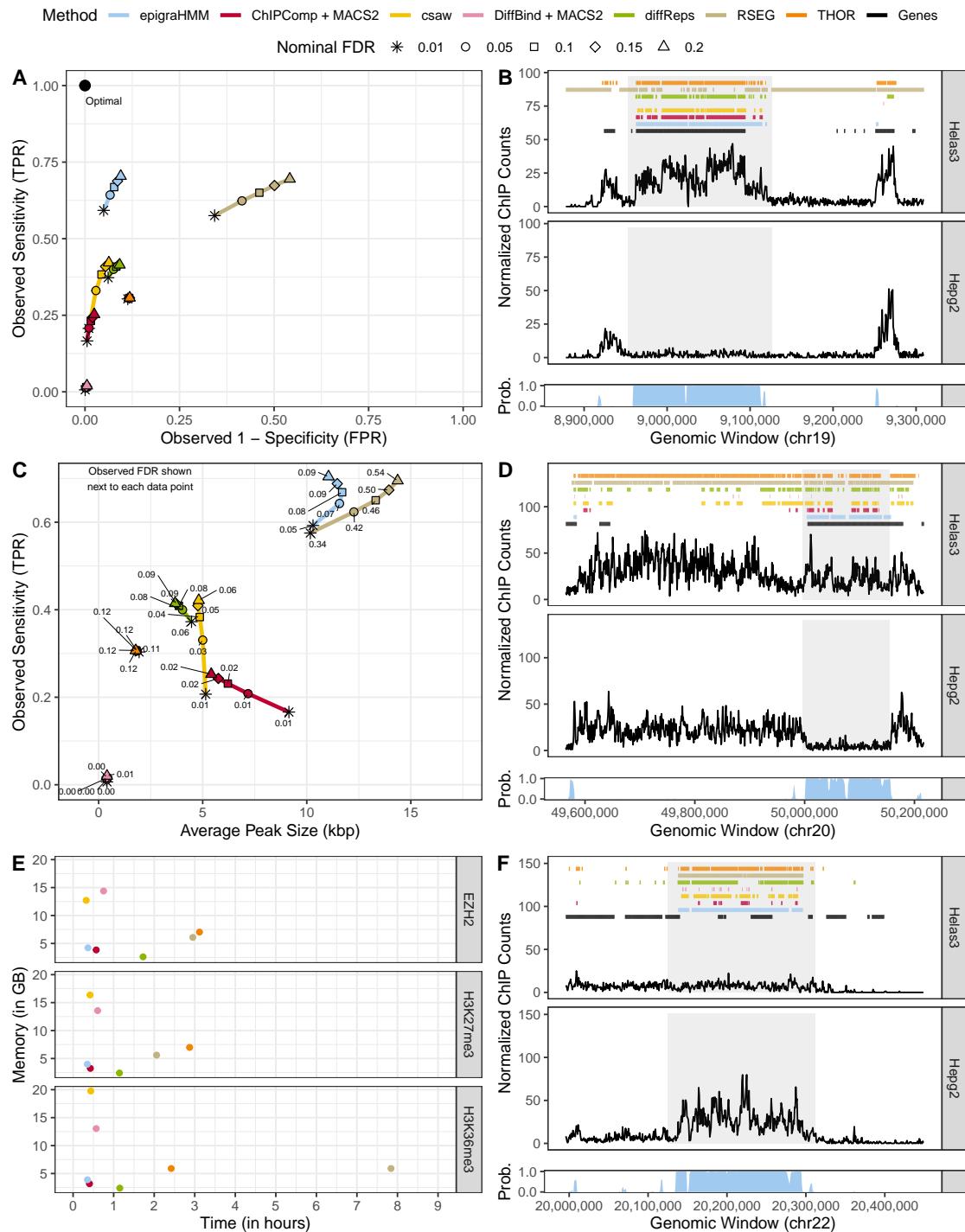
Web Figure 22: ROC curves for H3K36me3 differential peaks covering differentially transcribed gene bodies defined if having absolute $LFC > 1$ (A), $LFC > 1.5$ (B), $LFC > 2.5$ (C), or $LFC > 3$ (D).

D2 Study on Window Sizes

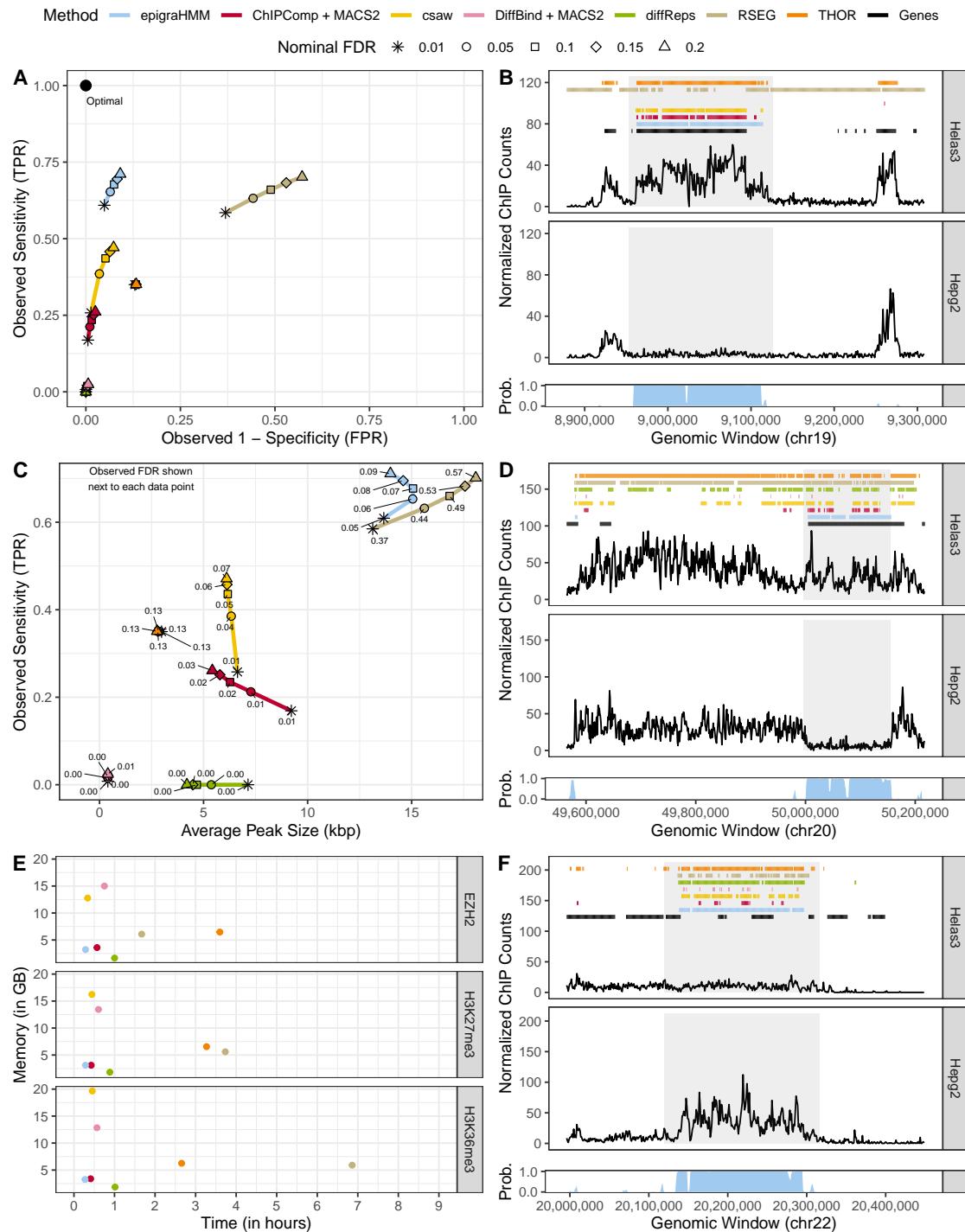
We compared methods regarding the metrics used in this paper to analyze broad marks using window sizes of 250bp, 500bp , 750bp, and 1000bp. Overall, results were consistent across different window sizes. The HMM-based methods RSEG and THOR appeared to be highly sensitive to the window size and showed better results for 1000bp. However, the presented model appeared to be robust to all scenarios as it consistently called a small number of broad peaks overall (Figures 23, 24, and 25, panels C). The methods csaw, ChIPComp, diffReps, and DiffBind exhibited fragmented and narrow differential peaks regardless of the window size. The performance of all methods appeared to be consistent to the choice of window size under sharp data set. The presented model was robust in all the scenarios as it was able to detect large differences between read counts of the analyzed cell lines.



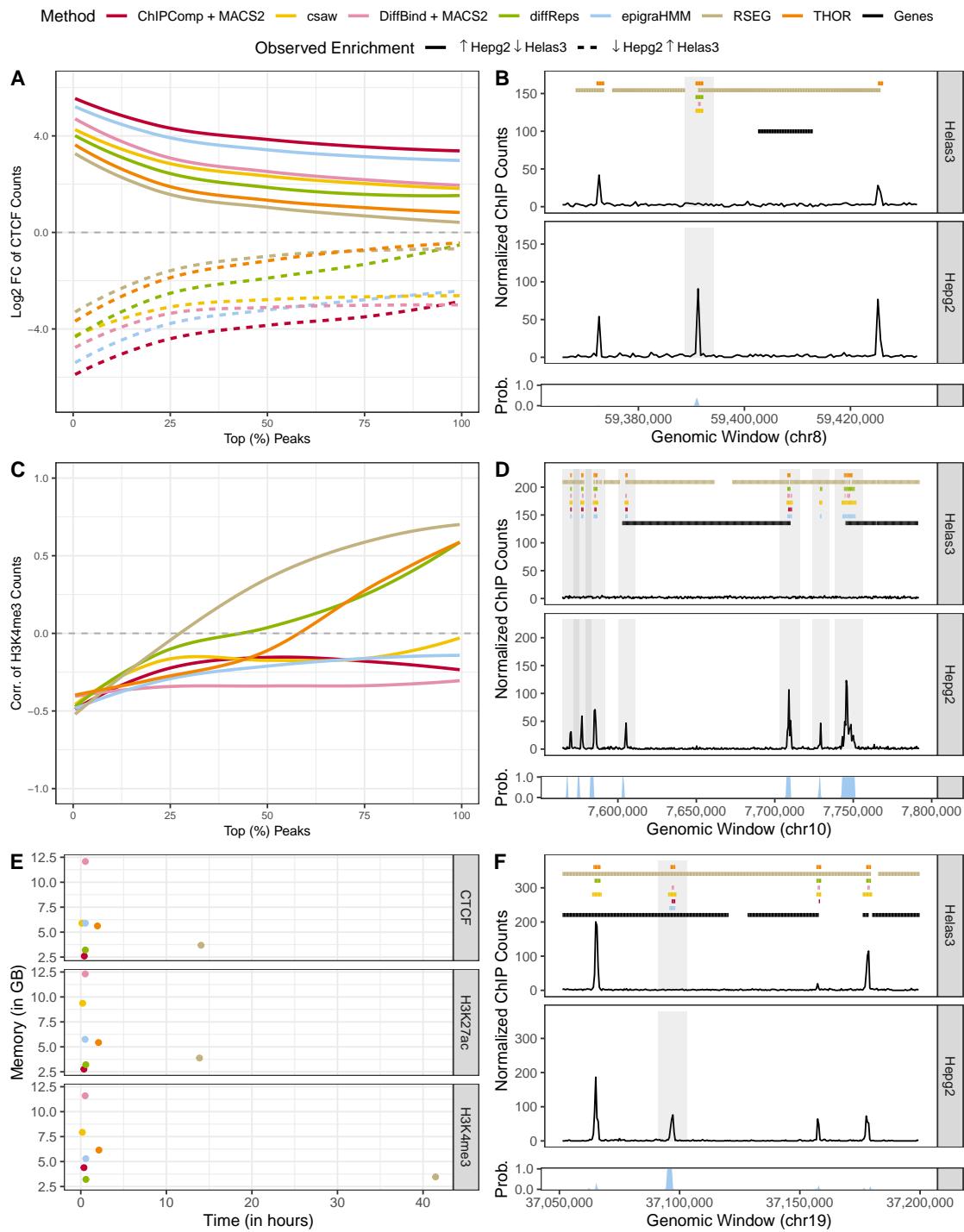
Web Figure 23: Results from broad marks (250bp). (A): ROC curves of differentially transcribed gene coverage based on H3K36me3 differential peak calls. (C): average number (y-axis) and size (x-axis) of H3K27me3 called peaks for various methods and different nominal FDR thresholds. (B), (D), and (F): example of peak calls from H3K36me3, H3K27me3, and EZH2, respectively. (E): computing time of genome-wide analysis of broad marks from various methods.



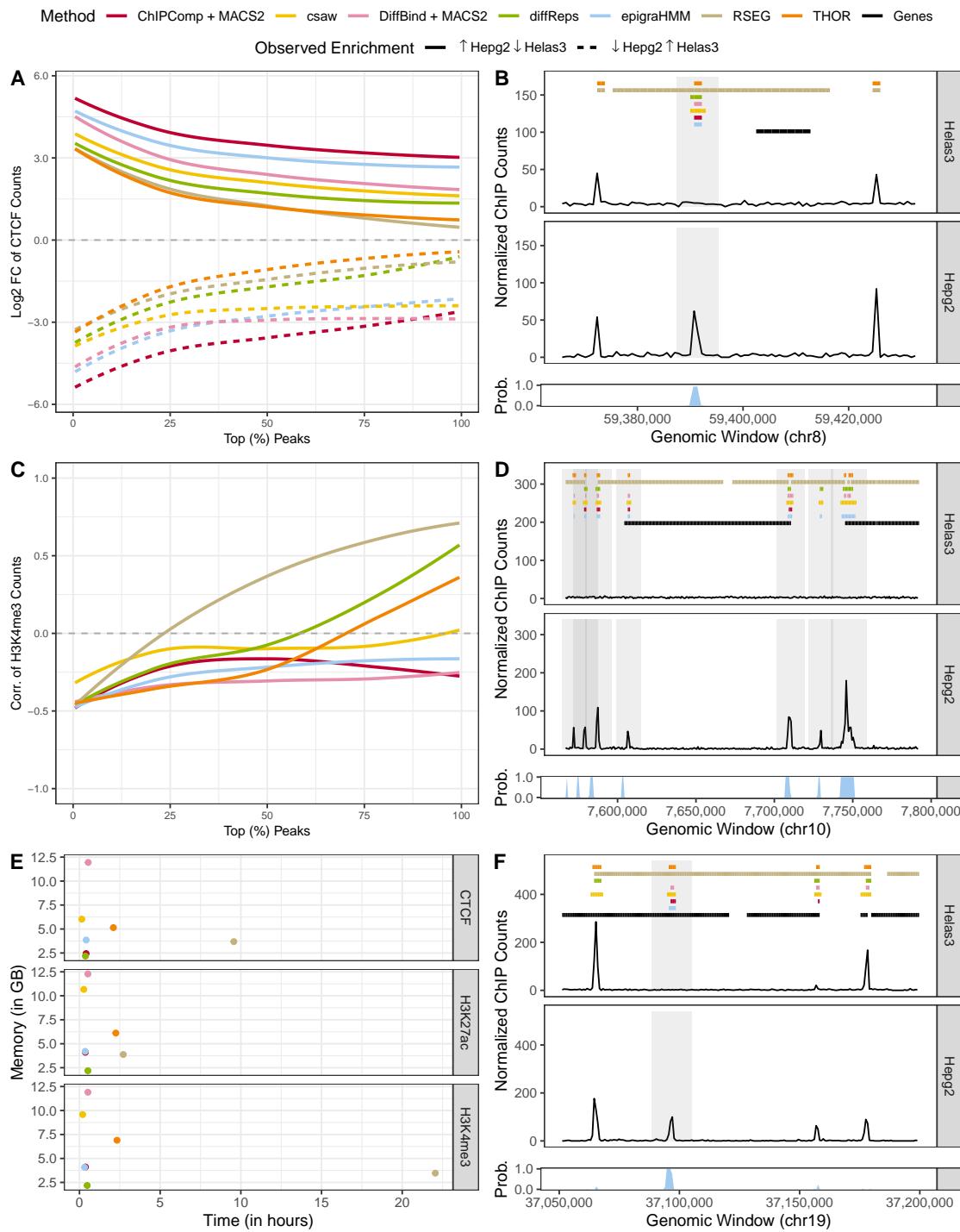
Web Figure 24: Results from broad marks (750bp). (A): ROC curves of differentially transcribed gene coverage based on H3K36me3 differential peak calls. (C): average number (y-axis) and size (x-axis) of H3K27me3 called peaks for various methods and different nominal FDR thresholds. (B), (D), and (F): example of peak calls from H3K36me3, H3K27me3, and EZH2, respectively. (E): computing time of genome-wide analysis of broad marks from various methods.



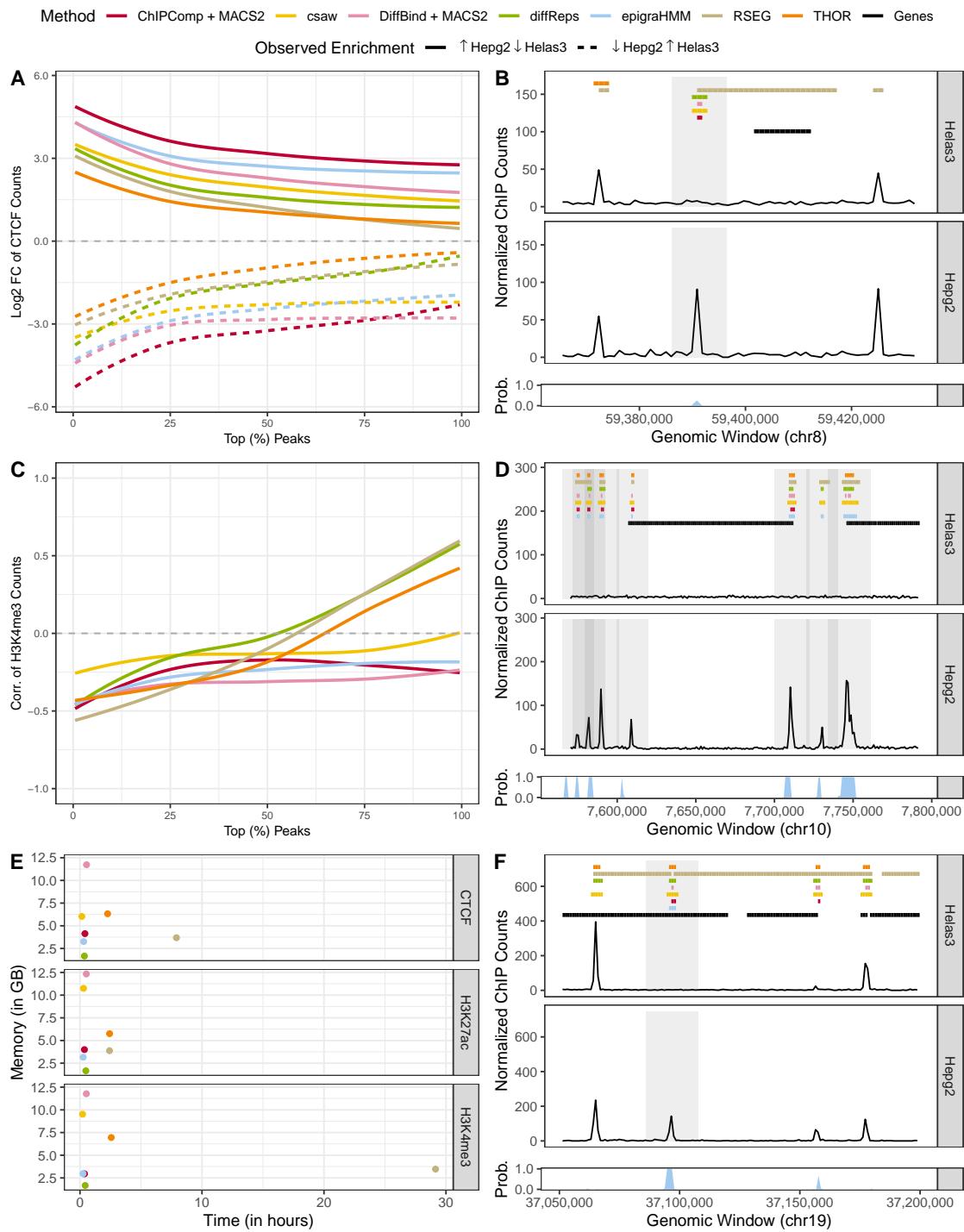
Web Figure 25: Results from broad marks (1000bp). (A): ROC curves of differentially transcribed gene coverage based on H3K36me3 differential peak calls. (C): average number (y-axis) and size (x-axis) of H3K27me3 called peaks for various methods and different nominal FDR thresholds. (B), (D), and (F): example of peak calls from H3K36me3, H3K27me3, and EZH2, respectively. (E): computing time of genome-wide analysis of broad marks from various methods.



Web Figure 26: Results from sharp marks (500bp). (A) and (C): median $\log_2 FC$ and correlation between cell lines of ChIP-seq counts from differential called peaks (FDR 0.05, sorted by the absolute LFC) for CTCF and H3K4me3, respectively. (B), (D), and (F): example of broad peak calls from CTCF, H3K4me3, and H3K27ac, respectively. (E): computing time of genome-wide analysis of sharp marks from various methods.



Web Figure 27: Results from sharp marks (750bp). (A) and (C): median log₂ FC and correlation between cell lines of ChIP-seq counts from differential called peaks (FDR 0.05, sorted by the absolute LFC) for CTCF and H3K4me3, respectively. (B), (D), and (F): example of broad peak calls from CTCF, H3K4me3, and H3K27ac, respectively. (E): computing time of genome-wide analysis of sharp marks from various methods.



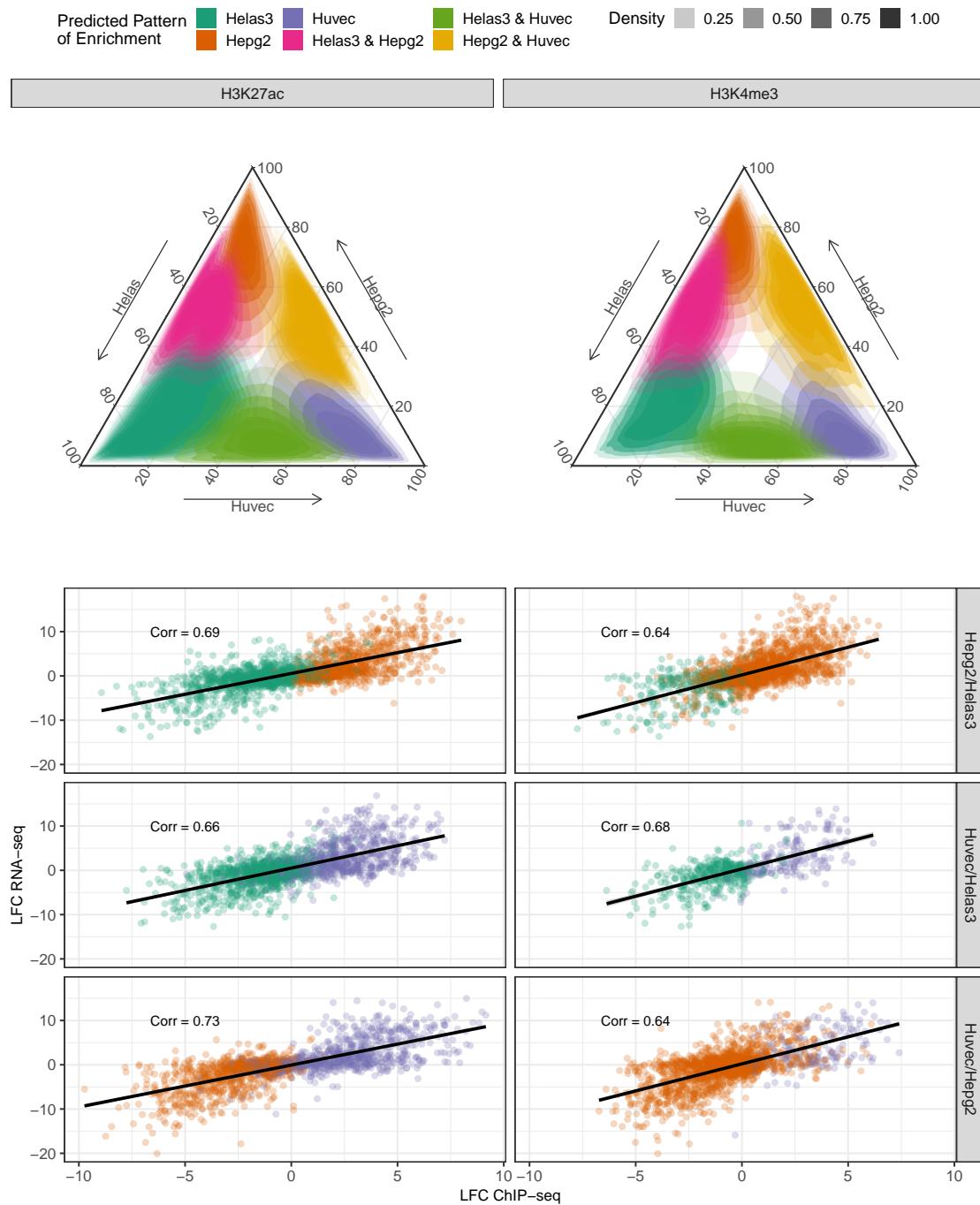
Web Figure 28: Results from sharp marks (1000bp). (A) and (C): median log₂ FC and correlation between cell lines of ChIP-seq counts from differential called peaks (FDR 0.05, sorted by the absolute LFC) for CTCF and H3K4me3, respectively. (B), (D), and (F): example of broad peak calls from CTCF, H3K4me3, and H3K27ac, respectively. (E): computing time of genome-wide analysis of sharp marks from various methods.

D3 Classification of Combinatorial Patterns of Protein-DNA Binding Activity and Their Associations With Gene Expression

Lastly, we jointly analyzed data from cell lines Helas3, Hepg2, and Huvec to assess the performance and biological relevance of the classification of differential combinatorial patterns of enrichment from H3K4me3 and H3K27ac. These epigenomic marks have been associated with gene transcription and are often deposited on promoter regions of actively transcribed gene bodies. First, we evaluated the classification performance of differential combinatorial patterns of enrichment across cell lines by making use of the estimated mixture model posterior probabilities. Given the differential HMM state, these posterior probabilities allow us assess whether read counts from genomic windows are likely to be generated from a particular combinatorial pattern of enrichment. Secondly, we compared the biological role of the classified patterns with gene expression data. To this end, RNA-seq and ChIP-seq read counts were mapped onto gene bodies and gene promoters, respectively. For the former, we used counts from abundance estimates scaled using the average transcript length over samples [12]. Using DESeq2 [6], normalization between samples was performed to avoid spurious differences due to sequencing depth in both RNA-seq and ChIP-seq data. Then, pairwise LFC of read counts between cell lines were calculated. Genes with zero counts or with low total read count across cell lines were excluded from the analysis (25% of the protein-coding genes). Finally, we used the maximum window-based posterior probability of the presented mixture model to classify the combinatorial patterns of enrichment from the resulting genes whose promoters overlapped differential peaks.

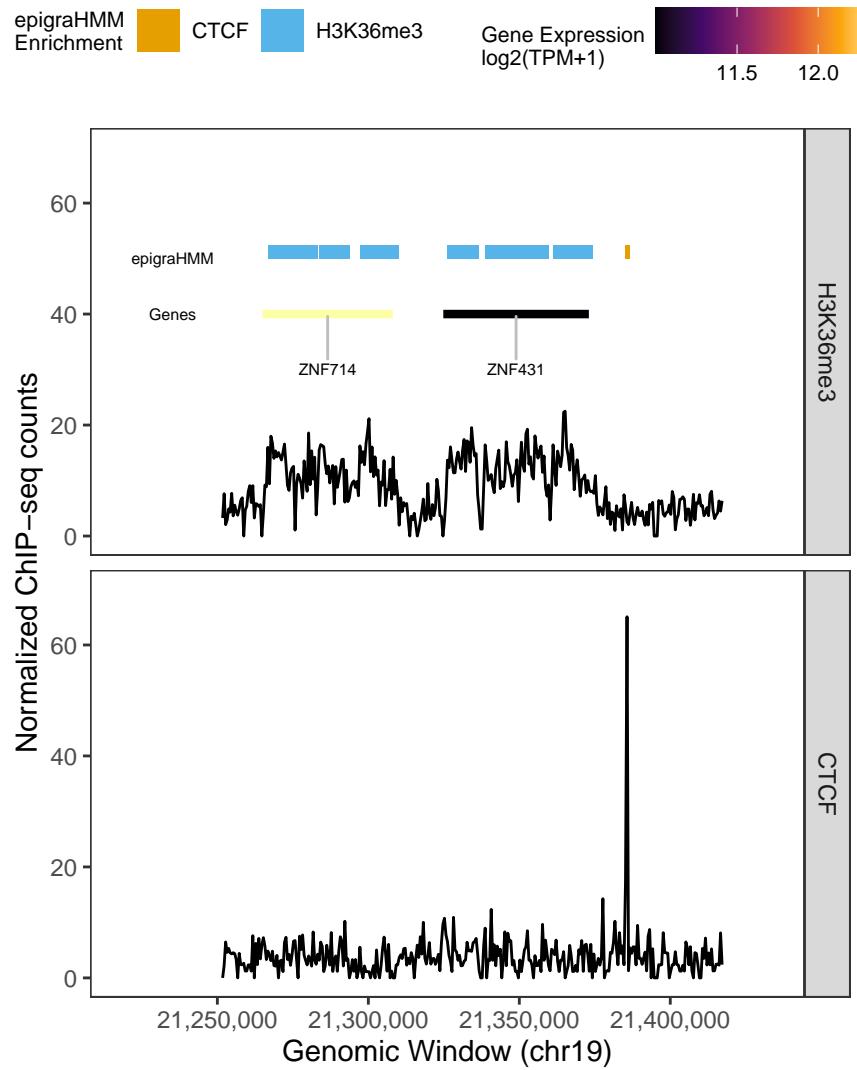
Figure 30 shows the main results from our analyses under a nominal FDR control of 0.05. At the top, we present ternary density plots of differential *peaks* and their associated ChIP-seq read counts. The colors indicate the classified differential pattern of enrichment of peaks based on window-based posterior probabilities of the mixture model. We observed that the

classification pattern agreed with the observed direction of enrichment from ChIP-seq data (vertices and opposite sides of the triangle vertices). At the bottom, we show the scatter plots of LFCs between cell lines of *gene* bodies (RNA-seq counts) and *gene* promoters (ChIP-seq counts), along with their Pearson correlation, for some of the classified combinatorial patterns of enrichment. As expected from the activating roles of H3K27ac and H3K4me3, we observed a high agreement between ChIP-seq and RNA-seq LFCs. Here, the colors indicate the classified differential pattern of activation of genes overlapping differential called peaks. These results show that the classified pattern of epigenomic change across the three cell lines agreed with the observed direction of gene expression and highlight the novelties of the presented model. We were able to accurately classify the differential pattern of enrichment of both called peaks (ternary plots) and genes overlapping called peaks (scatter plots) under more than two conditions. The embedded mixture model and its posterior probabilities generalizes the problem of differential peak calling for multiple conditions and provide a tool that is not yet available in any other method presented in the literature.



Web Figure 29: Ternary plots (top) of ChIP-seq counts mapped on differential peaks and scatter plots (bottom) of $\log_2 FCs$ (LFC) between cell lines of ChIP-seq (x-axis) and RNA-seq counts (y-axis) mapped on genes intersecting differential peaks. Colors indicate the classified combinatorial pattern of peak enrichment (ternary plots) and gene activation (scatter plots). Correlations indicate high agreement between ChIP-seq and RNA-seq $\log_2 FCs$ for these activating marks.

D4 Genomic Segmentation Analysis of H3K36me3 and CTCF



Web Figure 30: Genomic segmentation analysis of H3K36me3 and CTCF in Helas3 cell line. The chosen model parametrization and the normalization for non-linear biases via model offsets allow the segmentation of highly diverse epigenomic marks. The implemented hidden Markov model is able to properly account for the differences in length of enrichment regions between CTCF (short) and H3K36me3 (broad).

Bibliography

- [1] Manuel Allhoff, Kristin Seré, Juliana F. Pires, Martin Zenke, and Ivan G. Costa, *Differential peak calling of chip-seq signals with replicates with thor*, Nucleic acids research **44** (2016), no. 20, e153–e153.
- [2] Li Chen, Chi Wang, Zhaohui S Qin, and Hao Wu, *A novel statistical method for quantitative comparison of multiple ChIP-seq datasets*, Bioinformatics **31** (2015), no. 12, 1889–1896.
- [3] Jason Ernst and Manolis Kellis, *Chromhmm: automating chromatin-state discovery and characterization*, Nature methods **9** (2012), no. 3, 215–216.
- [4] Hongkai Ji, Xia Li, Qian-fei Wang, and Yang Ning, *Differential principal component analysis of chip-seq*, Proceedings of the National Academy of Sciences **110** (2013), no. 17, 6789–6794.
- [5] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, and Richard Durbin, *The sequence alignment/map format and SAMtools*, Bioinformatics **25** (2009), no. 16, 2078–2079.
- [6] Michael I Love, Wolfgang Huber, and Simon Anders, *Moderated estimation of fold change and dispersion for rna-seq data with deseq2*, Genome biology **15** (2014), no. 12, 550.

- [7] Aaron TL Lun and Gordon K Smyth, *csaw: a bioconductor package for differential binding analysis of chip-seq data using sliding windows*, Nucleic acids research **44** (2015), no. 5, e45–e45.
- [8] Alessandro Mammana and Johannes Helmuth, *bamsignals: Extract read count signals from bam files*, R package version **1** (2016), no. 3.
- [9] Rob Patro, Geet Duggal, Michael I Love, Rafael A Irizarry, and Carl Kingsford, *Salmon provides fast and bias-aware quantification of transcript expression*, Nature methods **14** (2017), no. 4, 417.
- [10] Aaron R Quinlan and Ira M Hall, *BEDTools: a flexible suite of utilities for comparing genomic features*, Bioinformatics **26** (2010), no. 6, 841–842.
- [11] Li Shen, Ning-Yi Shao, Xiaochuan Liu, Ian Maze, Jian Feng, and Eric J Nestler, *diffreps: detecting differential chromatin modification sites from chip-seq data with biological replicates*, PloS one **8** (2013), no. 6, e65598.
- [12] Charlotte Soneson, Michael I Love, and Mark D Robinson, *Differential analyses for rna-seq: transcript-level estimates improve gene-level inferences*, F1000Research **4** (2015).
- [13] Rory Stark and Gordon Brown, *Diffbind: differential binding analysis of chip-seq peak data*, R package version **100** (2011), 4–3.
- [14] Sebastian Steinhauser, Nils Kurzawa, Roland Eils, and Carl Herrmann, *A comprehensive comparison of tools for differential chip-seq analysis*, Briefings in bioinformatics **17** (2016), no. 6, 953–966.
- [15] Mingxiang Teng and Rafael A Irizarry, *Accounting for GC-content bias reduces systematic errors and batch effects in ChIP-seq data*, Genome research **27** (2017), no. 11, 1930–1938.

- [16] Walter Zucchini, Iain L MacDonald, and Roland Langrock, *Hidden markov models for time series: an introduction using r*, Chapman and Hall/CRC, 2017.