Journal of the American Statistical Association

ISSN: 0162-1459 (Print) 1537-274X (Online) Journal homepage: https://www.tandfonline.com/loi/uasa20

Template-Based Models for Genome-Wide Analysis of Next-Generation Sequencing Data at Base-Pair Resolution

Alexander W. Blocker & Edoardo M. Airoldi

To cite this article: Alexander W. Blocker & Edoardo M. Airoldi (2016) Template-Based Models for Genome-Wide Analysis of Next-Generation Sequencing Data at Base-Pair Resolution, Journal of the American Statistical Association, 111:515, 967-987, DOI: 10.1080/01621459.2016.1141095

To link to this article: https://doi.org/10.1080/01621459.2016.1141095

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2016, VOL. 111, NO. 515, 967–987, Applications and Case Studies
http://dx.doi.org/10.1080/01621459.2016.1141095

Taylor & Francis
Taylor & Francis Group

Template-Based Models for Genome-Wide Analysis of Next-Generation Sequencing Data at Base-Pair Resolution

Alexander W. Blocker and Edoardo M. Airoldi

Department of Statistics, Harvard University, Cambridge, MA, USA

ABSTRACT

We consider the problem of estimating the genome-wide distribution of nucleosome positions from paired-end sequencing data. We develop a modeling approach based on nonparametric templates to control for the variability along the sequence of read counts associated with nucleosomal DNA due to enzymatic digestion and other sample preparation steps, and we develop a calibrated Bayesian method to detect local concentrations of nucleosome positions. We also introduce a set of estimands that provides rich, interpretable summaries of nucleosome positioning. Inference is carried out via a distributed Hamiltonian Monte Carlo algorithm that can scale linearly with the length of the genome being analyzed. We provide MPI-based Python implementations of the proposed methods, stand-alone and on Amazon EC2, which can provide inferences on an entire *Saccharomyces cerevisiae* genome in less than 1 hr on EC2. We evaluate the accuracy and reproducibility of the inferences leveraging a factorially designed simulation study and experimental replicates. The template-based approach we develop here is also applicable to single-end sequencing data by using alternative sources of fragment length information, and to ordered and sequential data more generally. It provides a flexible and scalable alternative to mixture models, hidden Markov models, and Parzen-window methods. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received January 2013
Revised October 2015

KEYWORDS
Calibrated Bayesian detection; Deconvolution; Hamiltonian Monte Carlo; Massive data; Measurement error; Nucleosomes; Parallel computation; Yeast

1. Introduction

The organization of genetic material within cells plays a major role in the regulation of biological activities. In eukaryotic cells, DNA is wrapped around histone proteins to form nucleosomes, which constitute the smallest units of this organization. DNA must be accessible for transcription to occur; thus, the presence of nucleosomes physically constrains regulation. High-throughput sequencing technology produces indirect noisy evidence about the positions of nucleosomes across an entire genome, with an unprecedented resolution. In this article, we develop methods to provide accurate, reproducible estimates of nucleosome positions across a genome from high-throughput sequencing data, enabling the investigation of fine-grained structure in nucleosome positioning and its regulatory role.

We consider high-throughput sequencing data derived from micrococcal nuclease digestion (Tirosh 2012). Briefly, this technique involves linking histone proteins to the target DNA wrapped around them, digesting the remaining DNA using an enzyme, then digesting the histone proteins to make the target DNA accessible for further processing (e.g., see Tsankov et al. 2010). A gel is used to select DNA fragments with an approximate length of 150 base pairs—the length of DNA wrapped around each nucleosome. These fragments are amplified via PCR and sequenced (Albert et al. 2007). The resulting sequences, or reads, are aligned to a reference genome for the organism of interest using standard software (Bowtie; Langmead et al. 2009). The data consist of the number of read centers that align to each base pair along the genome. Figure 1 illustrates some example data.

We analyze data obtained with paired-end sequencing technology; that is, each DNA fragment is sequenced simultaneously from both ends, and the two reads are recorded as a pair. This technology provides the length of each fragment, in addition to its location, following alignment of the paired reads. These lengths are a valuable source of information about the digestion process, a major source of variation in the data. More recently, high-coverage paired-end experiments have become more affordable and biological research has shifted toward this technology (e.g., Gaffney et al. 2012). In previous work, the estimand of interest has consistently been the coarse-grained configuration of nonoverlapping nucleosomes for a population of cells. Advances in sequencing technology allow us to infer nucleosome positioning at a finer genomic scale, including stable nucleosomes that are alternatively positioned within subpopulations of cells—from aggregate population-level data. In this article, we propose a method for this inference using a combination of Bayesian and frequentist techniques.

1.1. Related Work

The positioning of nucleosomes along the genome was first studied with tiling microarrays (Yuan et al. 2005; Segal et al. 2006; Lee et al. 2007). High-throughput sequencing data allows

CONTACT Edoardo M. Airoldi airoldi@fas.harvard.edu Department of Statistics, Harvard University, 1 Oxford Street, Cambridge, MA 02138.
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/r/JASA
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JASA.

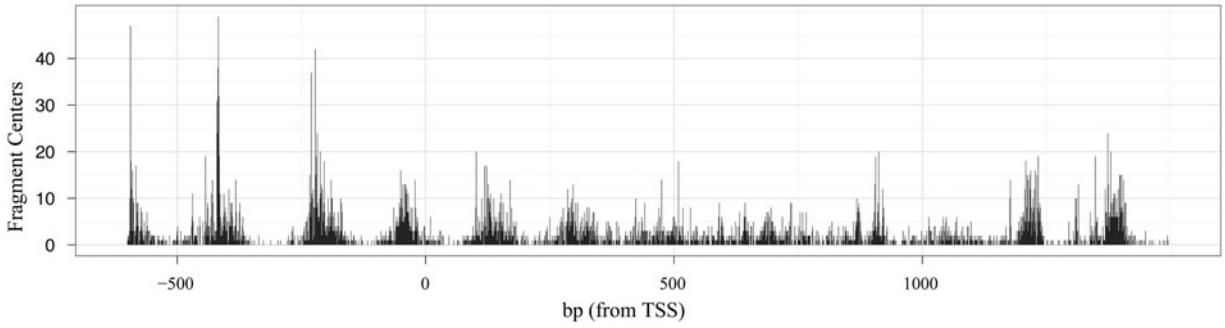

Figure 1. Example data for yeast gene PHO5.

for the analysis of nucleosome positioning in any organism, and overcomes many technical limitations of tiling microarrays (Jansen and Verstrepen 2011). The first wave of studies using high-throughput sequencing to infer nucleosome positioning used single-end sequencing technology (Albert et al. 2007; Shivaswamy et al. 2008; Tsankov et al. 2010). More recent studies have used paired-end sequencing technology (Gkikopoulos et al. 2011).

State-of-the-art statistical methods for identifying nucleosome positions from tiling microarray data are mostly based on hidden Markov models and variants thereof (Gupta 2007; Yassour et al. 2008; Yuan and Liu 2008; Sun et al. 2009b; Mitra and Gupta 2011). Mixture model-based approaches have also been fairly successful (e.g., Sun et al. 2009a). Most sequencing studies in the biological literature have used kernel density estimators, often referred to as Parzen-window based estimators. These convolve the observed read counts with a kernel, extract local maxima from the resulting density estimate, and compute other estimates by taking these maxima as nucleosome positions. Variants of this technique include the use of multiple windows (Weiner et al. 2010), frequency-based filtering using fast Fourier transformation (FFT) (Flores and Orozco 2011), and a Kolmogorov-Smirnov based method for detecting differences in nucleosome positioning between samples (Fu et al. 2012). Other studies have adapted HMMs to sequencing data (Cairns et al. 2011). Model-based analyses of sequencing data have focused on mixture models for relatively low-coverage datasets (Rashid et al. 2011; Polishko et al. 2012; Zhang et al. 2012). The methodology has recently shifted focus to multiresolution techniques and stochastic optimization for higher-coverage datasets (Polishko et al. 2014; Schöpflin et al. 2013, respectively).

A different strain of work combines a new biochemical protocol with a Bayesian deconvolution method (Brogaard et al. 2012). Their approach provides detailed information at a resolution of less than 10 bp, but it requires an intricate and nonstandard sample preparation that is hard to implement in the lab. We focus on the analysis of more common MNase-digested sequencing data, as that is the standard in the field.

Methods relevant to our work have been developed to identify binding sites from high-throughput ChIP-seq data (Park 2009). Analyses of ChIP-seq data often combine variants of Parzen-window estimation (Schwartzman et al. 2013), a Poisson model for sequence counts (Zhang et al. 2008), and detection methods for peak finding (Pepke, Wold, and Mortazavi 2009). Alternative approaches have included Ising models (Mo 2012). The combination of segmentation and permutation-based testing for local detection is also reminiscent of Frazee et al. (2014).

Methods such as Zhang et al. (2008), Barski and Zhao (2009), and Mo (2012) represent an interesting line of work but focus on a somewhat distinct problem from high-resolution nucleosome position estimation. They seek to identify binding regions, often characterized by motifs and other sequence features. These regions are nonoverlapping and are typically hundreds of base pairs in extent. Distances between identified peak centers are also typically large, on order of kilobases. This scale allows for a number of assumptions in models for ChIP-seq data (Barski and Zhao 2009) that cannot be defended in models of nucleosomes, since nucleosomes can be expected to overlap when the population of cells is sequenced. In contrast, our work is focused on the separation of nearby nucleosome centers with overlapping peak profiles in high-throughput sequencing data. The proximity of the positions of interest in our analysis (less than 10 base pairs) necessitates a stronger focus on digestion variation and a model that accommodates multiple nearby positions generating overlapping reads.

1.2. Contributions of this Article

We develop a *template-based approach* for estimating the genome-wide distribution of nucleosome positions from paired-end sequencing data. This approach uses information on fragment lengths provided by paired-end sequencing to estimate the amount of variation due to enzymatic digestion in each lane of sequencing data. Using this information, we posit a model that captures both the variation of read positions due to enzymatic digestion and the variation due other sources of experimental error, in Section 2. This model incorporates a hierarchical structure within discrete segments of the genome to provide local regularization. We also introduce a set of novel estimands that provide interpretable summaries of the genome-wide distribution of nucleosome positions.

We develop a parallel Hamiltonian Markov chain Monte Carlo (MCMC) sampler to draw from the posterior distribution of the quantities of interest under our model, in Section 3. This sampler is highly amenable to distributed computation and scales linearly with the length of the genome being analyzed. We provide a nonparametric estimator of the distribution of digestion errors and propose a segmentation algorithm that splits the genome in regions of similar coverage, respecting biological features. We introduce a calibrated Bayesian method

with frequentist error guarantees, to detect local concentrations of nucleosome positions.

We demonstrate the proposed methods on real and simulated data in Sections 4 and 5, assessing the accuracy and reproducibility of the inferences. We also compare the performance of our methods to the popular Parzen-window and read-based estimators.

2. Model

Here we develop a model for paired-end reads, obtained using Solexa high-throughput sequencing technology. The data consist of integer counts y_k of the fragment centers observed at each base pair k along an N-base pair long chromosome, together with the corresponding fragment lengths l_j for each of the M observed fragments, which provide information about how far apart the paired reads are.

The proposed model consists of two distinct components: an observation model $p(\boldsymbol{y}|\boldsymbol{\beta})$, which provides the distribution of the observed read counts given the underlying distribution of nucleosome positions $\boldsymbol{\beta}$, and a positioning model $p(\boldsymbol{\beta}|\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$, which describes the structure of the nucleosome position distribution. We take as given a segmentation function, $s : \{1 \ldots N\} \to \{1 \ldots S\}$, which maps the N base pair locations to S regions in which coefficients β_k can be assumed to be identically distributed. We posit

$$y_k|\lambda_k \sim \text{Poisson}\,(\lambda_k) \tag{1}$$

$$\boldsymbol{\lambda}_{(N\times 1)} \equiv X_{(N\times(N-2\lfloor \ell_0/2\rfloor))}\boldsymbol{\beta}_{((N-2\lfloor \ell_0/2\rfloor)\times 1)}, \quad \beta_k > 0 \text{ for } k = \lfloor \ell_0/2\rfloor + 1 \ldots N - \lfloor \ell_0/2\rfloor \tag{2}$$

$$\log \beta_k \sim \text{Normal}\,\left(\mu_{s_k}, \sigma^2_{s_k}\right), \tag{3}$$

where X specifies the contribution of a nucleosome positioned at base pair k to the expected number of reads at base pair m due to digestion variability, ℓ_0 is the base length of each fragment in the absence of digestion variation (typically 147 bp), and $s(k)$ is denoted as s_k for compactness. (The construction of the matrix X is detailed in Section 2.1.) The log-likelihood for the proposed model is as follows, subject to the positivity constraint on $\boldsymbol{\beta}$,

$$\begin{aligned}\log p(\boldsymbol{y}|\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) &= -\sum_k \boldsymbol{x}_k^T\boldsymbol{\beta} + \sum_k y_k \log\left(\boldsymbol{x}_k^T\boldsymbol{\beta}\right) \\ &\quad -\frac{1}{2}\sum_k \log\sigma^2_{s_k} - \frac{1}{2}\sum_k \frac{(\log\beta_k - \mu_{s_k})^2}{\sigma^2_{s_k}} + \text{const.}\end{aligned} \tag{4}$$

To complete the model specifications, we place priors on μ_s and σ_s^2. We use independent conjugate priors for σ_s^2, assuming $\sigma_s^2 \sim \text{InvGamma}(\alpha_0, \gamma_0)$. Our priors for μ_s are fully conjugate and independent across segments; we assume $p(\mu_s|\sigma_s^2) \sim N(\mu_0, \frac{\sigma_s^2}{n_s\tau_0})$ where n_s is the length of segment s. The dependence of the prior variance on n_s provides a consistent degree of regularization across segments of different lengths and was found to produce superior power and reproducibility. These stabilize our inferences and reflect vague prior information on the distribution of $\boldsymbol{\beta}$. This is particularly true for our prior on σ_s^2, which regulates the uniformity of nucleosome positioning. Their form also allows for efficient computation, as outlined in Section 3. We discuss the sensitivity of the inferences to the choice of $(\mu_0, \tau_0, \alpha_0, \gamma_0)$ in Sections 4 and 5.

The specified model is an effective working model for the desired analysis, particularly for the deconvolution of digestion variation, but it is not a complete mechanistic reflection of the underlying biology. The sequenced reads might include naked DNA, transient nucleosome positions, and other genomic features that we do not account for in the model. However, by allowing for "background" reads that do not reflect nucleosome positions of interest, the complete proposed procedure, consisting of cluster estimands and a detection procedure for local concentrations, is designed to be robust to such phenomena so long as the corresponding reads do not concentrate along the genome like true nucleosome reads. Even if the inferred β suffer from these issues and cannot be interpreted directly as a map of nucleosome positions, they remain useful as an intermediate step toward such a map.

The proposed model depends upon two technical constructs: digestion-variability templates and a segmentation of the DNA sequence. We discuss them further in the next two sections, before introducing the estimands of interest in Section 2.3.

2.1. Digestion Variability Template

A template summarizes variation due to enzymatic digestion in a single lane of sequencing and it is used to build the X matrix in Equation (2). We cover the paired-end case here and discuss extensions to single-end sequencing in Section 6. Consider a simple model for the variability of enzymatic digestion. We denote the length of each fragment j as ℓ_j and assume

$$\ell_j = \ell_0 + e_{1j} + e_{2j}, \quad e_{1j}, e_{2j} \sim \text{IID}. \tag{5}$$

ℓ_0 is the base length of each fragment, typically 147 bp, and the $e_{\cdot j}$ terms are the digestion errors at each end of the fragment. The error terms $e_{\cdot j}$ correspond to the number of base pairs by which a fragment of nucleosome-bound DNA is under-digested; negative values imply over-digestion. We assume these errors are bounded and symmetric between the ends of each fragment; physically, this means that the enzyme has the same propensity toward over-or under-digestion at each end. Under this model, each fragment's center varies about its nucleosome's true center according to the distribution of $d_j \equiv \frac{1}{2}(e_{1j} - e_{2j})$. Our template is this distribution, expressed in vector form and transformed to account for the random rounding of fragment centers to integer positions. Hence,

$$t_k = P(d_j = k) + \frac{1}{2}\left(P\left(d_j = k - \frac{1}{2}\right) + P\left(d_j = k + \frac{1}{2}\right)\right) \tag{6}$$

for $k = -w, \ldots, w$, yielding a vector $\boldsymbol{t}$ of length $2w + 1$. This equation follows directly from the assumption of randomized rounding, as noninteger digestion errors are mapped to either the preceding or succeeding integers with probability $1/2$, which mirrors our preprocessing.

We estimate the template from the empirical fragment length distribution corresponding to a lane of paired-end sequencing data, as detailed in Section 3.1. Example exact and approximate templates for the same data are shown in Figure 2.

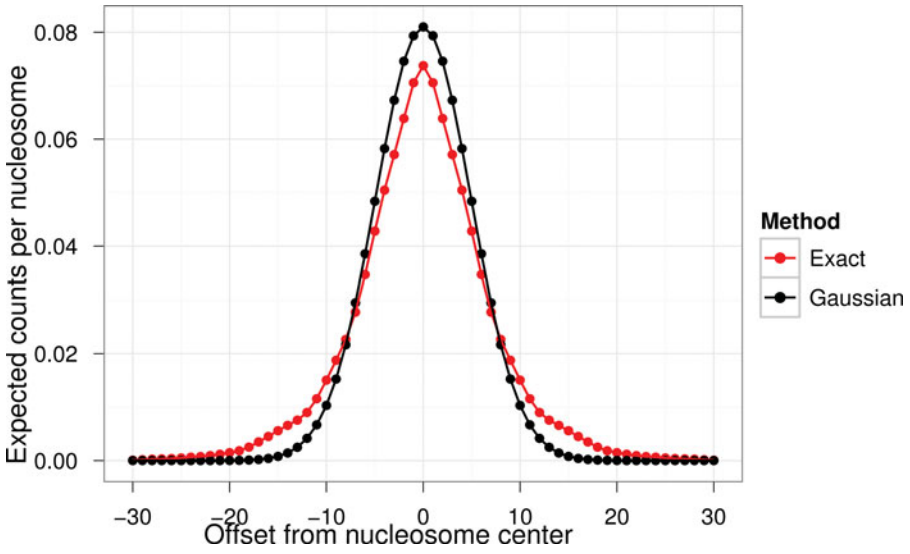

Figure 2. Example templates for yeast growing in high-phosphate.

The matrix X in Equation (2) is constructed using a template $\boldsymbol{t}$ by leveraging an equivalence between a realistic data-generating process and the marginal specification given in Equations (1) and (2). Briefly, an explicit model would combine a Poisson distribution for the unobservable number of reads that are generated from a given nucleosome location, with a multinomial distribution that controls the offsets of the observed read centers from the center of that nucleosome, which is where they would all be observed in the absence of digestion variability. This Poisson-multinomial structure for the observed reads is marginally equivalent to the more convenient Poisson GLM with an identity link function specified in Equations (1) and (2).

Denote the length of the sequence of interest N and the width of the template $2w + 1$ as above. Then we can define the digestion (or basis) matrix X as the $(N \times N - 2\lfloor \ell_0/2 \rfloor)$ convolution matrix for this template. Each row corresponds to a shifted version of the template, so X is banded and sparse with only $(2w + 1)N$ nonzero entries. For example, if $w = 1$ and $N = 7$, we would have

$$X = \begin{pmatrix} t_{-1} & & & & & \\ t_0 & t_{-1} & & & & \\ t_1 & t_0 & t_{-1} & & & \\ & t_1 & t_0 & t_{-1} & & \\ & & t_1 & t_0 & t_{-1} & \\ & & & t_1 & t_0 & \\ & & & & t_1 & \end{pmatrix}. \tag{7}$$

Using the $(N - 2\lfloor \ell_0/2 \rfloor)$-dimensional constrained vector of coefficients $\boldsymbol{\beta} \geq 0$, we obtain an N-dimensional vector of expected counts $\boldsymbol{\lambda}$ using Equation (2). Each coefficient β_k provides the number of fragments we expect to sequence from nucleosomes centered at position k. Analogously, $\boldsymbol{\lambda}$ provides the number of fragment centers we expect to observe at each position. Formally, $\boldsymbol{\lambda}$ is a convolution of $\boldsymbol{\beta}$ with $\boldsymbol{t}$. This structure models the effect of digestion variability on the observations.

Digestion variability affects the statistical properties of our data in two important ways under this model. First, the expected counts of fragment centers are convolved with the digestion variability template. This reduces the concentration of counts at each nucleosome position, obscuring the true center of the nucleosome. Second, as digestion variability convolves the expected fragment counts over a broader stretch of the genome, the expected number of counts at each base pair decreases, driving down the signal-to-noise ratio. This phenomenon is not unique to Poisson noise, but it is particularly acute in this setting because the signal-to-noise ratio of a Poisson random variable is equal to its expectation. The combination of these effects makes inferring nucleosome positions very challenging in this setting, even in high-coverage experiments. The resulting combination of "vertical" noise (from Poisson-log-normal variation) and "horizontal" convolution across the sequence (from digestion variability) creates a challenging deconvolution problem.

2.2. Segmentation

Our model uses a segmentation of the DNA sequence to account for variation in occupancy, coverage, and structure. The goal is to split chromosomes into local regions where the IID assumption on the coefficients β_k is sufficient to provide effective regularization. The segmentation function s defined above must fulfill a monotonicity condition, $s(k+1) - s(k) \in \{0, 1\}$, so that segments are indexed contiguously and in strictly increasing order. An example segmentation of yeast chromosome I is shown in Figure 3.

Statistically, the segmentation enables local regularization in the estimation of $\boldsymbol{\beta}$. These coefficients are weakly identified in a model specified by Equations (1) and (2) alone. Such a model would involve $N - 2\lfloor \ell_0/2 \rfloor$ parameters and N observations, and the Hessian matrix for the implied log-likelihood of $\boldsymbol{\beta}$ would be $H = -X^T W X$, where $W = \text{diag}\left(y_1/\boldsymbol{\lambda}_1^2, \ldots, y_n/\boldsymbol{\lambda}_n^2\right)$. This is negative-definite if $\boldsymbol{y}$ contains no all-zero subvector of length $2w + 1$ or more; otherwise, it is only negative semidefinite. Furthermore, H is typically ill-conditioned due to the convolution structure of X. Estimates of β_k from this model would be extremely unstable. We regularize the estimates of β_k by modeling the distribution of nucleosome positions with Equation (3). In this complete model, we pool information locally within each

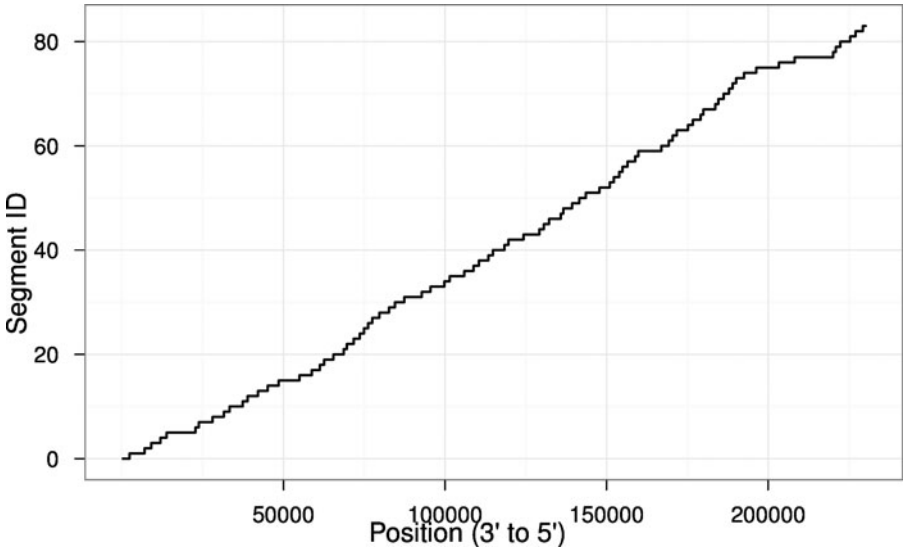

Figure 3. Example segmentation of yeast chromosome I. See Section 3 for estimation details.

chromosome, as β_j is independent of β_k if $s_j \neq s_k$, where s_k is the segment to which base pair k belongs.

Segments divide each chromosome into local stretches over which a consistent distribution of nucleosome positions is plausible. We posit a log-normal distribution for the magnitudes of the coefficients β_k. The idea is that most locations on the sequence are expected to have a very low concentration of nucleosome positions. These locations correspond to small, nonzero values of β_k. A few locations have a relatively high concentration of nucleosomes across a population of cells. These are the positions of interest, corresponding to large values of β_k. The log-normal distribution captures this behavior: it allows the majority of values in $\boldsymbol{\beta}$ to concentrate around a low baseline rate with a few values many orders of magnitude larger than the baseline. The parameters μ_{s_k} and $\sigma^2_{s_k}$ control this baseline and the prevalence of extreme values in $\boldsymbol{\beta}$, providing us with a flexible, parsimonious way to regularize our estimation and provide more reliable inferences.

The segmentation also provides a way to control the bias-variance trade-off of our regularization. Using a large number of short segments results in low bias, as they can capture sequence features at a fine scale. This also leads to greater uncertainty, as more parameters are introduced and less observations are available for regularization within each segment. Using a smaller number of longer segments produces the opposite effect. We discuss one strategy to manage this trade-off in Section 3.2.

2.3. Estimands

We can express the scientific estimands of interest as functions of $\boldsymbol{\beta}$. The parameter $\boldsymbol{\beta}$ itself is of interest, as it captures the pattern of nucleosome positioning across each chromosome. However, $\boldsymbol{\beta}$ is high-dimensional and difficult to interpret directly. The posterior expectation, standard deviation, and quantiles of $\boldsymbol{\beta}$ are useful for visualization and exploratory analysis, but answering scientific questions about nucleosome positioning requires more targeted measures. Below, we introduce more refined estimands useful in quantifying the structure in the nucleosome position distribution. These estimands fall into two broad categories: (1) local measures of concentration, and (2) cluster-level summaries of structure.

The first family of estimands aims to quantify the relative concentration of nucleosome centers within a local window. Formally, for each base pair location k in $\boldsymbol{\beta}$, we consider the ratio

$$C_{p,l}(k) = \frac{\sum_{i=-p}^{p} \beta_{k+i}}{\sum_{i=-l}^{l} \beta_{l+i}}, \tag{8}$$

where $2l+1$ is the width of a local window and $2p+1$ is the width of the region of interest. We typically choose $l = 73$, yielding a local window of width 147. For $l \leq 73$, the structure of $\boldsymbol{\beta}$ within $2l+1$ bp windows can be taken as measure of the distribution of nucleosome positions across the population of cells. Physically, a nucleosome consists of 147 bp of DNA wrapped around histone proteins, so, within a single cell, nucleosomes must be spaced by at least 147 bp. As a result, each cell can contribute at most one nucleosome center within a window of width 147 bp or less. Thus, the relative magnitudes of the entries of $\boldsymbol{\beta}$ within such a window reflect only the distribution of nucleosome positions across cells, not the arrangement of multiple nucleosomes within any individual cell.

Choosing $p = 0$ yields a measure of relative concentration at each base pair in the chromosome. Selecting $p > 0$ typically produces more scientifically relevant estimands as it allows for a bit of biological variation in nucleosome positions. These estimands come with a useful baseline. Assuming a uniform local distribution of nucleosome positions across cells in the population would imply $C_{p,l}(k) = \frac{(2p+1)}{(2l+1)}$. Deviations from this baseline provide a normalized measure of local concentration. We present strategies for the detection of local nucleosome concentrations based on these estimands in Section 3.4.

This measure is analogous to the positioning score presented by Gaffney et al. (2012) and similar in concept to the stringency score of Valouev et al. (2008). They differ formally because the proposed estimands $C_{p,l}(k)$ are defined on the parameters β_k, whereas the positioning score of Gaffney et al. (2012) is defined on the read midpoints themselves. This allows our measure to capture finer-grained structure, as we show in Sections 4 and

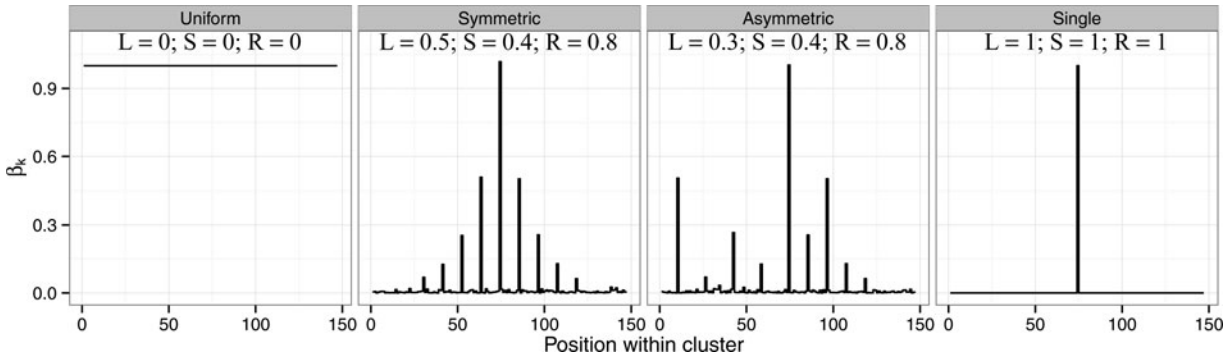

Figure 4. Examples of cluster estimands for four cases: uniform, symmetric with low-frequency positions, asymmetric with low-frequency positions, and a single position. All estimands have $i = 1$ and $j = 147$, and $q = 0.9$ for R.

5. We obtain highly reproducible results for p as low as 1 bp, compared to the 15 bp of Gaffney et al. (2012).

The second family of estimands provides summaries of small clusters of nucleosome positions. By definition, these estimands rely on a procedure to identify local clusters (e.g., Parzen-window filtering) applied to the vector of coefficients $\boldsymbol{\beta}$. These estimands will inherit any shortcomings of the clustering procedure they rely on. This is not a major issue for comparisons across datasets, but some caution is needed in their interpretation. We define the estimand $\boldsymbol{\kappa}$ to be the cluster centers obtained by running the selected clustering method on $\boldsymbol{\beta}$. $\boldsymbol{\kappa}$ is a cluster-level estimand itself, but it is primarily of interest as a means to obtain summaries of $\boldsymbol{\beta}$ within individual clusters. We consider measures of structure, localization, and sparsity within the cluster, defined as the normalized entropy, mean absolute deviation, and quantiles of the entries of $\boldsymbol{\beta}$, taken as an unnormalized discrete distribution over the base pairs in the cluster. Formally, considering cluster $\boldsymbol{\beta}_{[i:j]}$ and defining $p_{[i,j]}(k) = \beta_k / \sum_{m=i}^{j} \beta_m$, the proposed localization, structure, and sparsity estimands are defined as

$$
\begin{aligned}
L_{i,j} &= 1 - \frac{4}{j-i+1} \sum_{k=i}^{j} p_{[i,j]}(k)|k - m_{i,j}|, \\
m_{i,j} &= \sum_{k=i}^{j} k p_{[i,j]}(k) \qquad (9) \\
S_{i,j} &= 1 + \frac{1}{\log(j-i+1)} \sum_{k=i}^{j} p_{[i,j]}(k) \log p_{[i,j]}(k) \qquad (10) \\
R_{i,j,q} &= 1 - \frac{n_{i,j,q} - 1}{q(j-i+1)}, \\
n_{i,j,q} &= \min\left(n : \sum_{k=i}^{i+n} \tilde{p}_{[i,j]}(k) \geq q \right), \qquad (11)
\end{aligned}
$$

respectively, where $\tilde{p}_{[i,j]}(k)$ is $p_{[i,j]}(k)$ sorted in descending order. All measures are normalized so $L_{i,j} = S_{i,j} = R_{i,j,q} = 0$ if $\boldsymbol{\beta}_{[i:j]}$ is constant and $L_{i,j} = S_{i,j} = R_{i,j,q} = 1$ if $\boldsymbol{\beta}_{[i:j]}$ contains only one nonzero entry.

The proposed localization measure $L_{i,j}$ is similar in motivation to the stability score defined by Shivaswamy et al. (2008), capturing the variability of nucleosome positions within each cluster. The proposed sparsity and structure measures capture different features of nucleosome positioning within clusters. Structure ($S_{i,j}$) provides a position-invariant measure of uniformity, and sparsity ($R_{i,j,q}$) captures the concentration of nucleosomes at multiple, possibly noncentral positions. Biologically, the localization measure answers the question "how far from the center are nucleosomes in the average cell?" The structure measure answers the question "how nonuniform is the distribution of nucleosome positions across cells?", and the sparsity measure answers the question "how many distinct positions account for $q \cdot 100\%$ of cells?" Sparsity and structure measures answer similar questions in different ways; the framing of the sparsity measure is often more interpretable for biologists, whereas the structure measure is less interpretable but requires no selection of q.

Figure 4 shows these estimates for four possible distributions of nucleosome positions within a 147-bp cluster. All estimands are zero in the uniform case and 1 in the single position case by design. The symmetric and asymmetric cases have the same marginal distribution of β_k, so the structure and sparsity estimands are identical. The localization estimand is larger in the asymmetric case as frequent alternative positions are farther from the center. These cases show how our estimands separate order-invariant properties of the nucleosome distribution from those that depend on the exact locations of nucleosome concentrations. These cluster-level estimands allow us to quantify differences like this in a simpler, lower-dimensional way than the local concentration estimands can provide.

Using the methods described in Section 3, we can obtain draws from the posterior distribution of all of these estimands. This allows us to cleanly separate the modeling of the measurement process and broad properties of nucleosome positioning from the features of interest. Using estimates for the parameters β_k instead of the raw counts y_k allows us to reliably estimate these quantities, as we show in Section 5.

3. Inference, Estimation, and Computation

To extract useful inferences from the model of Section 2, we must address three sets of unknown quantities: the digestion template $\boldsymbol{t}$, the segmentation of each chromosome s, and the parameters and latent variables of the positioning model, $\boldsymbol{\beta}$, $\boldsymbol{\mu}$, and $\boldsymbol{\sigma}^2$. The parameter $\boldsymbol{\beta}$ and quantities derived from it are of the greatest scientific interest, as they correspond directly to the chromatin structure. Before inferring $\boldsymbol{\beta}$, we address $\boldsymbol{t}$ and s using separate sources of information.

To estimate the template t from paired-end sequencing data, we develop a nonparametric method, in Section 3.1. We develop a simple algorithm to segment each chromosome into nonoverlapping segments with useful biological and statistical properties, in Section 3.2. Using the estimated template t and segmentation s, we turn to model-based inference for $(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$. We build a parallel MCMC algorithm that can efficiently sample from the joint posterior of these parameters, in Section 3.3. By combining the conditional independence structure of our model with distributed computation, we are able to handle datasets where $\boldsymbol{\beta}$ contains millions of entries.

An approximate EM algorithm is also provided in the online supplementary materials as an optional initialization step for this sampling. The EM approach provides a computationally efficient way to obtain rough estimates of these parameters, but the joint posterior distribution of $(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ has a complex multimodal structure that EM is ill-equipped to address. Implementation details are given in the online supplementary materials.

Finally, we calibrate the frequentist operating characteristics of our Bayesian estimators using a permutation null hypothesis, detailed in Section 3.4. This ensures that our conclusions are valid as Bayesian posterior probability assessments under our working and under frequentist criteria. We focus on the latter for our final scientific inferences, controlling the false discovery rate (FDR) for the detection of local structure in the distribution of nucleosome positions.

3.1. Template Estimation

Recall from Section 2.1 that we model the length of each fragment as $l_j = l_0 + e_{1,j} + e_{2,j}$. We assume that $e_{1,j}$ and $e_{2,j}$ (the digestion errors) are independent and identically distributed, and l_0 is fixed at 147, which is the known length of DNA wrapped around a single nucleosome. Figure 5 illustrates how $e_{1,j}$ and $e_{2,j}$ relate to each fragment. Along the genome, the distributions of digestion errors at the ends of each fragment are mirror images of each other, so positive values imply that some DNA not bound to a nucleosome is not digested and negative values imply over-digestion. Independence across digestion errors, symmetry between the endpoints of each fragment, and homogeneity of this distribution across the genome are the only modeling assumptions we make about the digestion process in the template model.

To set up the corresponding estimation problem, we define two probability distributions,

$$p(i) = \Pr(l_j = i) \tag{12}$$

$$q(i) = \Pr(e_{1,j} = i). \tag{13}$$

Physically, $l_j \geq 0$, so it is reasonable to assume that $e_{1,j}, e_{2,j} \geq -\lfloor \frac{l_0}{2} \rfloor$; any smaller values would allow for negative-length fragments with positive probability. Analogously, if the longest observed fragment length is $l_{\max}$, we have $\hat{Pr}\left(l_j > l_{\max}\right) = 0$. We therefore have $l_j \leq l_{\max}$, which implies $e_{1,j}, e_{2,j} \leq l_{\max} - l_0 + \lfloor \frac{l_0}{2} \rfloor$ for our nonparametric MLE. Thus, we can write

$$p(i) = \sum_{k=-\lfloor \frac{l_0}{2} \rfloor}^{l_{\max}-l_0+\lfloor \frac{l_0}{2} \rfloor} q(k)q(i - l_0 - k). \tag{14}$$

The resulting log-likelihood for the observed fragment lengths is

$$\ell(q) = \sum_{j=1}^{M} \log p(l_j). \tag{15}$$

We maximize this log-likelihood numerically, using a multivariate logit transformation on the values of $q(k)$ to avoid bounded optimization. Using the L-BFGS algorithm (Zhu et al. 1997) with numerical derivatives on a laptop with a Core i5 processor and 8 GB of RAM, this maximization requires approximately 40 sec for a typical experiment. As we take the multivariate logit transformation of $q(k)$, no box constraints are necessary. This computation scales only with the number of unique fragment lengths observed, so it does not constitute a bottleneck.

We obtain the template distribution t from q via a convolution sum and linear transformation. These calculations follow directly from the model assumed in Section 2.1; no further approximations or assumptions are required. Recall from

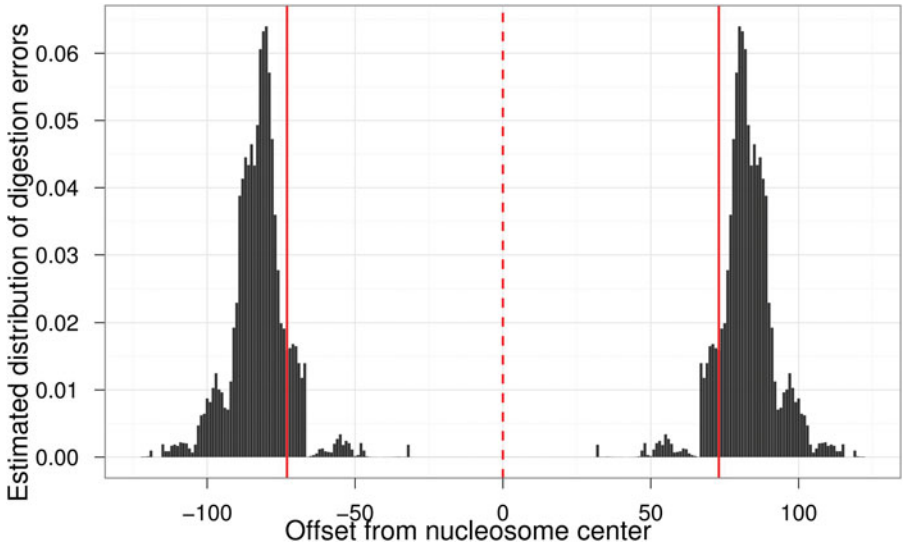

Figure 5. Estimated digestion error distributions versus offset from nucleosome center; vertical lines at $\pm l_0/2$ (solid) and nucleosome center (dashed).

Section 2.1 that t is the distribution of $\frac{e_1-e_2}{2}$, restricted to the integers via random rounding. We first obtain the distribution of $e_1 - e_2$ via

$$P(e_1 - e_2 = i) = \sum_{k=-\lfloor \frac{l_0}{2} \rfloor}^{l_{\max}-l_0+\lfloor \frac{l_0}{2} \rfloor} q(k)q(k-i). \tag{16}$$

This follows from the assumption of independence between e_1 and e_2. We finally transform the distribution of $e_1 - e_2$ to the desired template $t(i)$ by accounting for random rounding, as

$$t(k) = \frac{1}{2}P(e_1 - e_2 = 2k-1) + P(e_1 - e_2 = 2k) + \frac{1}{2}P(e_1 - e_2 = 2k+1). \tag{17}$$

The same random rounding procedure is used to map fragment centers to a single location on the genome. Thus, the estimated template accurately reflects both variation due to enzymatic digestion and the details of the preprocessing. We use the template estimated with this procedure to build the design matrix X in the observation model, as discussed in Section 2, and for the simulation study discussed in Section 4.1.

In MNase-seq experiments, it is common to restrict reads to a small set of lengths Λ near 147 base pairs before analysis to limit the impact of digestion variation. If such selection is performed, then the full set of fragment lengths without selection must be used to estimate the digestion error distribution $q(k)$. The template $t(k)$ corresponding to the restricted set of reads can then be obtained by calculating

$$P(e_1 - e_2 = i) = \frac{\sum_{k=-\lfloor \frac{l_0}{2} \rfloor}^{l_{\max}-l_0+\lfloor \frac{l_0}{2} \rfloor} q(k)q(k-i)\mathbf{1}_{(\ell_0+2k-i)\in\Lambda}}{\sum_{k=-\lfloor \frac{l_0}{2} \rfloor}^{l_{\max}-l_0+\lfloor \frac{l_0}{2} \rfloor} \mathbf{1}_{(\ell_0+2k-i)\in\Lambda}}. \tag{18}$$

If the restricted set of reads is instead used to estimate the digestion error distribution, then digestion variation is severely underestimated.

3.2. Segmentation Algorithm

We estimate the segmentation function $s : \{1 \dots N\} \to \{1 \dots S\}$ by leveraging the biological structure of each chromosome. We begin by enumerating all open reading frames (ORFs) and intergenic regions on a given chromosome. Merging overlapping ORFs into single segments yields a starting set of contiguous, nonoverlapping segments. Many of these segments are too short to provide useful local regularization. To increase the segmentation's utility, we merge neighboring segments until all segments exceed a minimal length (800 bp for the purposes of the analysis in Section 5).

We iteratively merge the most similar short segments until the resulting segmentation fulfills the given minimum length constraint. We measure similarity using the coverage within each segment, defined as

$$c_i = \frac{1}{n_i} \sum_{k:s(k)=i} y_k \,, \tag{19}$$

where n_i is the length of segment i. Algorithm 1 provides pseudocode for this procedure.

Given Minimum segment length M; initial segmentation;
Calculate $\{n_i\}$ and $\{c_i\}$;
while $\min_i n_i < M$ **do**
  Clear minimal difference in coverages d_m and index i_m;
  `/* Find the best merge among short segments */`
  for $i : n_i < M$ **do**
    Compute $d_i = \min(|c_i - c_{i-1}|, |c_i - c_{i+1}|)$;
    if $d_i < d_m$ **then**
      Update $d_m = d_i$ and $i_m = i$;
  `/* Execute best merge */`
  Merge segment i_m with neighbor having nearest coverage;
  Update $\{n_i\}$ and $\{c_i\}$;
return *Segmentation s*

Algorithm 1: Segmentation algorithm

At the conclusion of Algorithm 1, we obtain a segmentation for which each segment has enough observations to provide useful local regularization. The boundaries of each segment also align with biologically meaningful features, as every step in the above algorithm maintains segment boundaries as a subset of ORF boundaries. This segmentation is estimated once, and then used in all subsequent inference.

This segmentation strategy is effective for simple organisms such as *Saccharomyces cerevisiae*, but we would not recommend it, although it is feasible, for more complex genomes, such as mammals. Complications including alternative splicing, long intergenic regions, and a profusion of regulatory regions necessitate a more detailed analysis in complex organisms. However, the segmentation step is simply a preinference estimation module, which provides the regularization structure for the Hamiltonian Monte Carlo (HMC) sampler described in Section 3.3, and it can be easily replaced. For example, Frazee et al. (2014) used a hidden Markov model for segmentation, and segmentation by histone state is another plausible alternative. Any segmentation method could be substituted into the overall procedure with no other modifications.

3.3. Distributed Hamiltonian Monte Carlo Sampler

The MCMC sampler consists of two alternating updates. At each iteration r, our algorithm

1. Draws $(\boldsymbol{\mu}^{(r)}, \boldsymbol{\sigma}^{2\,(r)})|\boldsymbol{\beta}^{(r-1)}$ directly, then
2. Updates $\boldsymbol{\beta}^{(r)}|(\boldsymbol{\mu}^{(r)}, \boldsymbol{\sigma}^{2\,(r)})$ via a distributed HMC step.

The first update is straightforward as we can directly sample from the conditional posterior of $(\boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{2\,(t)})$. This is a standard conjugate normal update, given the log-normal hierarchical structure, and operates independently across segments. We give details in the online supplementary materials.

The second update is computationally challenging. The chromosomes of *S. cerevisiae* range in length from 230,218 to 1,531,933 base pairs, so the $\boldsymbol{\beta}$ vectors are very high-dimensional. In some of the experiments discussed in Section 4, we work with simulated chromosomes with over 3.85 million base pairs.

The conditional posterior of $\boldsymbol{\beta}^{(t)}|(\boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{2\,(t)})$ is not part of any standard family, so we turn to HMC. The dimensionality of $\boldsymbol{\beta}$ makes a single HMC update for the entire vector computationally infeasible and numerically unstable. To enable fast, statistically efficient computation, we take advantage of the conditional independence structure of this conditional posterior.

Subvectors of $\boldsymbol{\beta}$ separated by at least $2w$ entries are conditionally independent given $(\boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{2\,(t)})$ and the entries of $\boldsymbol{\beta}$ between them. Consider the subvectors $\boldsymbol{\beta}_{[j_1:j_2]}$ and $\boldsymbol{\beta}_{[k_1:k_2]}$, with $j_1 < j_2 < k_1 < k_2$. The elements of $\boldsymbol{\beta}_{[j_1:j_2]}$ affect only $\boldsymbol{\lambda}_{[j_1-w:j_2+w]}$, and the elements of $\boldsymbol{\beta}_{[k_1:k_2]}$ affect only $\boldsymbol{\lambda}_{[k_1-w:k_2+w]}$. Hence, if $k_1 > j_2 + 2w$, then $\boldsymbol{\beta}_{[j_1:j_2]}$ and $\boldsymbol{\beta}_{[k_1:k_2]}$ are conditionally independent given $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$.

We take advantage of this conditional independence to construct a distributed set of HMC updates. We first fix the length of each subvector that will be updated via a single HMC step to $B > 4w$. Next, consider two partitions of $\boldsymbol{\beta}$ into subvectors. The first starts at the beginning of $\boldsymbol{\beta}$ and proceeds forward with subvectors of length at most B separated by $2w$, yielding

$$\boldsymbol{\beta}_{[1:B]}, \boldsymbol{\beta}_{[B+2w+1:2B+2w]}, \ldots, \boldsymbol{\beta}_{[n_b(B+2w)+1:N]}\,.$$

The second begins at the $B/2$th entry of $\boldsymbol{\beta}$ and again proceeds forward in subvectors of length at most B, as

$$\boldsymbol{\beta}_{[B/2+1:3B/2]}, \boldsymbol{\beta}_{[3B/2+2w+1:5B/2+2w]}, \ldots, \boldsymbol{\beta}_{[n_b(B+2w)B/2+1:N]}\,.$$

Within each partition, the subvectors are conditionally independent, and, in combination, these partitions include all entries of $\boldsymbol{\beta}$.

Within each iteration of our sampler, we cycle through each of these partitions, updating each subvector of $\boldsymbol{\beta}$ with one HMC step. As each subvector within each partition is conditionally independent, we can execute all HMC steps in parallel for each partition. This allows us to distribute the computational burden over hundreds of cores, providing fast scalable inference. Each of these distributed HMC steps is, on its own, relatively standard. They are much less computationally intense than one would expect for a typical regression problem, as the log-conditional posterior's value and gradient can both be computed via a convolution, lowering the computational cost per core to $O(B \log B)$ with the fast Fourier transform. In particular, all matrix-vector products involving the X matrix can be computed as convolutions with the template vector $\boldsymbol{t}$ instead, reducing the complexity of these operations from $O(B^2)$ to $O(B \log B)$. Details of distributed algorithm, computational infrastructure, and tuning of the HMC are given in the online supplementary materials. A Python implementation of the sampler is available on GitHub; *http://www.github.com/airoldilab/cplate*.

3.4. Detection and Calibration

Recall from Section 2.3 that we quantify local concentrations of nucleosomes using the estimand $C_{p,l}(k)$, which defines local concentrations as small regions of the chromosome that contain a density of nucleosomes greater than that we would expect under a uniform distribution of nucleosomes across cells in our population, p/l.

We can estimate $P(C_{p,l}(k) > (2p+1)/(2l+1) \mid \boldsymbol{y})$ for each base pair k using the MCMC sampler described in Section 3. While these are useful, we demand stronger guarantees from our detection procedure than the Bayesian approach alone can provide. We want external validity under perturbations of the data, not simply internal validity given the posited model. To quantify the operating characteristics of our procedure and provide frequentist guarantees on its performance, we turn to permutation nulls.

One null hypothesis is that $\boldsymbol{y}$ consists of a set of multinomial draws. Under this null, the entries of $\boldsymbol{y}$ within each segment i are drawn from a multinomial distribution with equal probability assigned to each base pair within the segment and $n = \sum_{k:s(k)=i} y_k$. This null hypothesis rests on the same idea as Fisher's exact test: we condition on the marginal distribution of the data and consider all independent permutations of the observations. We approximate this null distribution by repeatedly randomly permuting the observed reads within each segment.

Another null hypothesis is that fragment positions are distributed at random within each segment, subject to MNase digestion bias. Under this null, we can draw from the null distribution of fragment midpoints in two stages. First, we draw from a multinomial distribution with probabilities corresponding to the observed distribution of cut dinucleotides to obtain the number of fragments with each pair of cut dinucleotides within a given segment. Second, we randomly assign fragments to locations within the segment conditional on the sampled number of counts. This assures that the resulting distribution of cut dinucleotides matches that observed in the original data.

With the results of these null simulations in-hand, we then run our MCMC sampler on each draw from the null. Under both nulls, we can reuse the segmentation estimated from the observed data. Under the digestion-biased null, we must reestimate the digestion template on the new data as cut dinucleotide distribution-preserving resampling is not guaranteed to preserve the exact length distribution. With the uniform null, we reuse the original template. From the sampler's output, we obtain an estimate of the distribution of $P(C_{p,l}(k) > (2p+1)/(2l+1)|\boldsymbol{y})$ over positions k under the null. We compare this to the distribution of posterior probabilities for the observed data and set a detection threshold to control the FDR using the method of Storey and Tibshirani (2003). For example, with the datasets analyzed in Section 5, we have typically found that a threshold of approximately 0.8 on $P(C_{p,l}(k) > (2p+1)/(2l+1)|\boldsymbol{y})$ yields an FDR of 5% or less. This approach provides a stringent detection procedure with Bayesian and frequentist validity, similar in spirit to the procedure of Frazee et al. (2014). Bayesian detection procedure such as the one developed by Newton et al. (2004) could be used in place of this permutation-based approach; only a single set of posterior samples would be required, but the frequentist guarantees of the proposed procedure would be lost.

4. Simulation Results

Here, we demonstrate the proposed methods on simulated data. The simulation studies aim to demonstrate the utility of the estimands introduced in Section 2.3 used in combination with the proposed deconvolution approach to inference, as well as the scalability of our methods. In Section 4.1, we describe the design of the simulation studies. We simulate high-throughput sequencing data with different coverage, on genes with primary

and alternative nucleosome positions, with different degrees of variation throughout the population. In Section 4.2.1, we compare the performance of the proposed method to that of a Parzen-window estimator followed by greedy search (the standard in the field; Albert et al. 2007; Shivaswamy et al. 2008; Tsankov et al. 2010; Tirosh 2012) for estimating measures of structure, quantifying power and error in estimating the locations of clusters of nucleosome positions. In Section 4.2.2, we assess the performance of the proposed method for estimating measures of local concentration, quantifying power and error in estimating the locations of distinct primary and alternative positions. In both Sections 4.2.1 and 4.2.2, we use design-based analysis of variance (ANOVA) to quantify the relative contributions to estimation errors of coverage, distance between primary and alternative positions and their relative frequencies across the cell population. We also use logistic regression to analyze the sensitivity of power to the three experimental factors we consider.

To perform inference throughout this section, we set $\mu_0 = 0$, $\tau_0 = 1/10$, $\alpha_0 = 7$, and $\gamma_0 = 10$. These values were chosen to be weakly informative on the basis of prior biological information. These values of α_0 and γ_0 imply that there is a 99% prior probability that 0.2% –13% of base pairs have β_k greater than or equal to 10 times their median. We found little sensitivity of our inferences to these choices of parameter values on a set of analyses on chromosome I. Varying τ_0 over two orders of magnitude (0.01–1) showed little effect on inferences, as did similar changes to (α_0, γ_0). Since the simulations were designed to be informative about the expected performance on real data, and to represent as many features of real data as possible, the prior settings above are the same as those used in Section 5.

4.1. Experimental Design

To assess the performance of the proposed methods, we generated artificial chromosomes using the classical principles of experimental design. These artificial chromosomes consist of a series of genes, each containing a set of nucleosome positions. We fix the length of each gene to 3501 bp, consisting of a 1000 bp promoter region, of a 2500 bp coding region, and of a 1 bp transcription start site (TSS).

We designed a simulation with three factors, varied at the gene level: coverage (the expected number of reads per gene), the spacing between primary nucleosome positions and alternative positions (which we refer to as offset), and the relative magnitudes of primary and alternative positions. Coverage had 10 levels, spanning the 5th–95th percentile observed gene-level coverages in increments of 10%. Alternative position spacing had 10 levels, spanning from 0 bp (no alternative positions) to 45 bp in increments of 5 bp. Alternative position magnitude had 11 levels, spanning from 0 (no alternative positions) to 1 (alternative positions of the same magnitude as primary positions) in increments of 0.1. Thus, the effective magnitude of the primary position relative to the alternative positions ranged from 1 to $\frac{1}{3}$. We used a full factorial design on these three factors, yielding 1100 distinct treatments for each of 10 simulated chromosomes. We then constructed a realistic distribution of nucleosome positions within each artificial gene. Using one of our high-phosphate datasets, we first identified clusters of nucleosome positions using the standard Parzen-window method. We indexed these clusters by their ordering within each actual gene, considering 1000 bp before TSS to the end of each ORF, and computed the proportion of reads within the ORF observed within each cluster. Finally, we averaged over the positions and proportions of these clusters by their order from their TSS, obtaining the average offset from the TSS and relative occupancy of the first, second, third, etc. clusters before and after the TSS. Figure 6 provides an illustration of coefficients, β_k, and read counts y_k, for this representative gene.

To generate our artificial dataset, we followed a modified version of the generative process outlined in Section 2. For each gene, we first drew coefficients for its subset of $\boldsymbol{\beta}$ from an upper-truncated log-normal distributed with parameters estimated from those regions with similar coverage. These are our "background" positions. Then, we set the entries of $\boldsymbol{\beta}$ corresponding to the gene's primary and alternative positions deterministically. The sum of the coefficients for these positions was fixed to the remaining total occupancy of the gene, less the sum of the background positions. Their relative magnitudes were determined by the design described above, with two alternative positions placed symmetrically around each primary position at

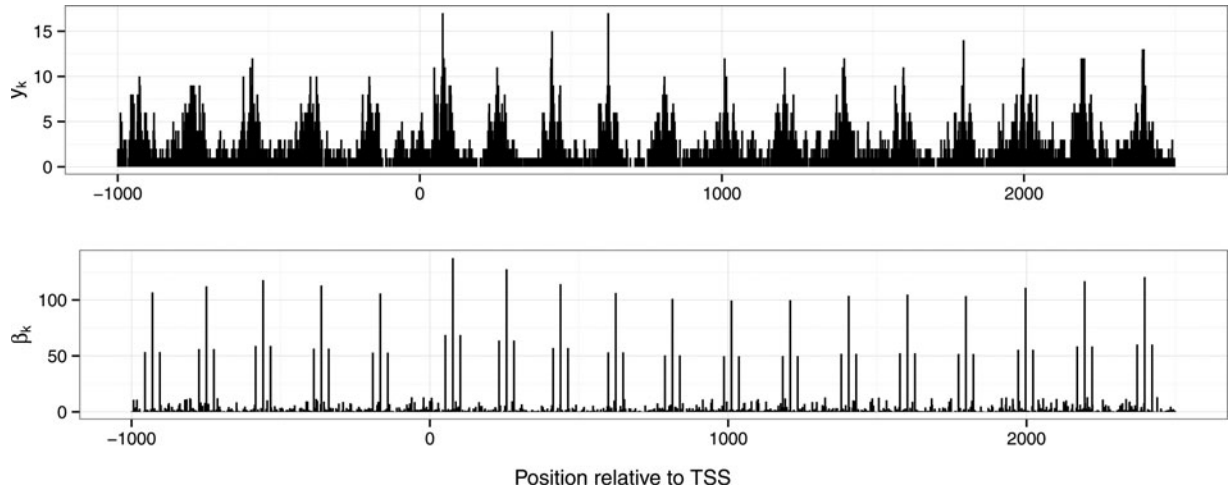

Figure 6. Illustration of one simulated gene: 0.55 quantile of coverage, with alternative position magnitude of 0.5, and alternative positions at ±25 bp from each primary position. Read counts y_k (top panel), and coefficients β_k (bottom panel).

the designated spacings. Thus, for a given level of coverage, the expected number of reads within each cluster was fixed, but its distribution across primary and alternative positions varies.

We convolved these $\boldsymbol{\beta}$ vectors with the template estimated from the experimental data to obtain vectors of expected read counts $\boldsymbol{\lambda}$. Finally, we generated $\boldsymbol{y} \sim$ iid Poisson $(\boldsymbol{\lambda})$ to obtain simulated read counts. This entire procedure was repeated for each replicate, yielding 10 artificial chromosomes of length 3,851,100 bp each.

These simulations are inspired by our generative model, but the data are not generated from the proposed model. We use the proposed model only for simulating the distribution of "background" coefficients and the effects of digestion. We introduce additional structure into the locations of nucleosome concentration via the deterministic placement of primary and alternative positions. This provides a more stringent test of our methods and decreases the residual variation across our experimental replicates. It also reflects our view of the generative model as a working approximation rather than a literal description of the underlying processes. As a "sanity check" on this design, simulated read counts were shown side-by-side with matched actual read counts to experienced biologists in this field. They could not reliably distinguish between the simulated and actual data. We provide further algorithmic details for this procedure and supporting figures in the online supplementary materials.

4.2. Power and Errors

Using simulated chromosomes, we compare the performance of the proposed method to that of a Parzen-window estimator (as used in Albert et al. 2007; Valouev et al. 2008; Tsankov et al. 2010; Tirosh 2012) and NOrMAL (Polishko et al. 2012) for estimating locations of clusters of nucleosome positions. We considered comparing against template filtering (Weiner et al. 2010) as well, but decided against it based on the extensive experiments of Polishko et al. (2012), which showed that NOrMAL provides greater power for detecting nucleosome clusters. We also assess the ability of the proposed method to detect and estimate the locations of primary and alternative positions. For both analyses, we use ANOVA to quantify the relative contributions to estimation errors of coverage, distance between primary and secondary positions and their relative frequencies across the cell population. We complement these with logistic regression to analyze the sensitivity of power to these three experimental factors.

The ground truth consists of the primary and alternative positions generated in Section 4.1, along with their coefficients. Recall that the output of the calibrated detection procedure, detailed in Section 3.4, is a series of positions where high local concentrations have been detected, and the output of the cluster-based estimands is a series of cluster centers. To assess performance in estimating cluster positions, we match inferred cluster centers to ground truth cluster centers. Similarly, to assess performance of local concentration measures, we match detected local concentrations to all ground truth positions, primary and alternative. The distance between each ground truth position and the nearest estimated position measures the power of a method without regard for false positives. Large distances imply low power, and vice versa. Conversely, finding the nearest ground truth position for each inferred position yields measures of precisions. For a perfect method, both distances would be zero as each true position would match to exactly one estimated position 0 bp away. Calling a concentration or cluster at every position would lead to a good "power" statistic, as every true position would be corresponding exactly to an estimated one, but a poor "precision" statistic, as most estimated positions, would be far from true ones.

The analyses below are based on summaries of these matched distances; we compute mean and median absolute errors, and we tabulate the proportion of true positions matched to an inferred positions within a fixed number of base pairs. The first set of quantities summarize distributions of errors in estimated positions, while the second is a measure of power.

4.2.1. Clusters

Detection of a cluster was defined as a best-match distance of less than 5 bp between the inferred and true cluster center (κ_m). Figure 7 summarizes our key findings, showing the power of each method as a function of the effective magnitude of the primary position, the offset of the alternative positions, and gene-level coverage. Table 1 provides the results of a design-based ANOVA of the mean absolute errors of estimated cluster locations, by gene, and Table 2 provides the results of a logistic regression of power on the design factors.

For estimating cluster locations, the proposed method dominates the Parzen-window estimator in power, with average difference of 2.1%, and mean absolute error (not shown) across all conditions. Power ranges from approximately 12% to 100% over all factor combinations in our experiments for both methods, while mean absolute position errors range from approximately 0.1 bp to 60 bp. Our method provided an average power of 85%, while the Parzen-window method's average power was 83%. Power shows a strong dependence on the local distribution of nucleosome positions; the accuracy in identifying the cluster centers of primary positions is reduced by the presence of stable alternative positions. The spacing between primary and alternative positions also affects power substantially, with power diminishing by approximately 30% as the offset increase from 0 bp to 35 bp. Power increases slightly for both methods at offsets of 40 bp and 45 bp. The relative performance of the proposed method is largest for offsets over 30 bp, with a difference in power of 7% at 45 bp. We observe little marginal dependence upon local coverage with only a 6% change in power over the range of coverage for both methods.

NOrMAL performs poorly throughout the design space. This is not surprising, as NOrMAL was designed to identify nonoverlapping nucleosome configurations from much lower-coverage data. Thus, while NOrMAL does not appear appropriate in this case, it may have utility in other settings.

The ANOVA and logistic regression analyses provide further insights into the role of interactions between the design factors. ANOVA results in Table 1 indicate that alternative position offset, effective magnitude, and their interaction account for the majority of variation in absolute position errors (approximately 75% of total variation and 94% of explained variation) for both methods. Logistic regression results in Table 2 suggest that

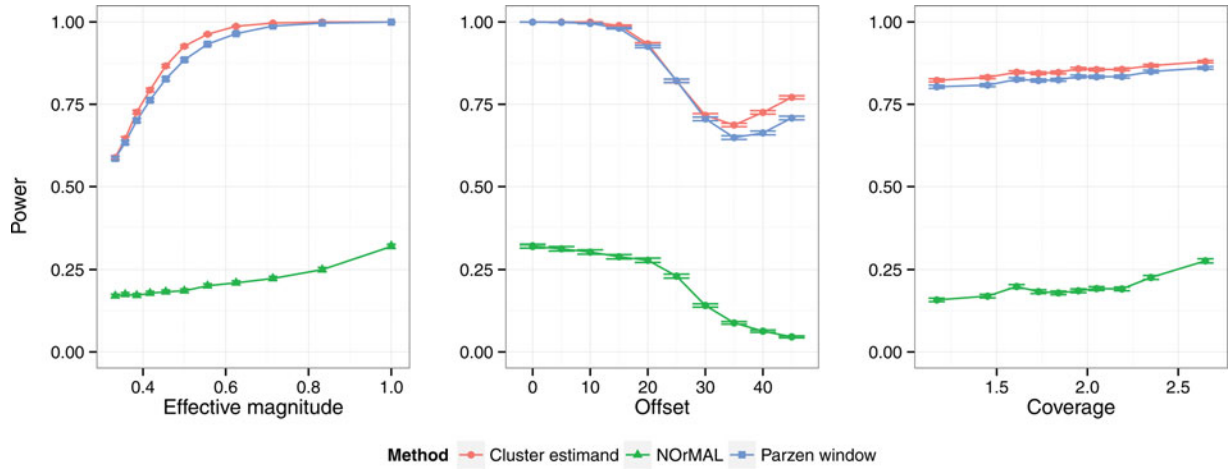

Figure 7. Power versus effective magnitude (left), alternative position offset (center), and coverage (right) for Parzen-window and cluster-estimand methods.

Table 1. Analysis of variance of absolute errors in cluster centers for cluster-estimand and Parzen-window methods.

		Cluster estimand		Parzen window		NOrMAL	
	Df	Sum Sq	*F* value	Sum Sq	*F* value	Sum Sq	*F* value
Coverage	9	1928.94	198.73	753.24	79.04	1546.48	61.40
Offset	9	19422.58	2001.01	10789.06	1132.10	23496.64	932.89
Magnitude	9	10764.55	1109.02	6768.51	710.22	2049.35	81.37
Coverage:Offset	81	275.41	3.15	264.68	3.09	307.25	1.36†
Coverage:Magnitude	81	175.19	2.01	142.80	1.66	235.54	1.04†
Offset:Magnitude	72	23535.40	303.09	19172.33	251.47	11885.99	58.99
Coverage:Offset:Magnitude	648	950.25	1.36	947.80	1.38	2992.97	1.65
Residuals	10090	10881.93		10684.31		28237.38	

NOTE: All factors and interactions except for those denoted with † were statistically significant with $p < 0.0001$.

Table 2. Logistic regression of power on design factors as continuous variables cluster estimand and Parzen window method.

	Cluster estimand		Parzen window		NOrMAL	
	Estimate	*z* value	Estimate	*z* value	Estimate	*z* value
(Intercept)	3.3461	54.15	3.4660	56.74	− 0.4025	− 11.04
Coverage	0.6069	5.20	0.7498	6.54	0.8868	14.29
Offset	− 5.3115	− 58.00	− 5.4596	− 60.92	− 3.4171	− 42.19
Effective Magnitude	2.7211	10.76	2.3296	10.82	− 0.6292	− 13.66
Coverage·Offset	− 0.4217	− 2.48	− 0.6071	− 3.66	− 0.2003	− 1.48
Coverage·Effective magnitude	1.7927	3.20	0.9037	2.02	− 0.3470	− 4.41
Offset·Effective magnitude	8.9842	22.27	6.9932	21.15	3.4500	25.22
Coverage·Offset·Effective magnitude	2.1476	2.52	1.8962	2.84	− 0.0859	− 0.37

NOTE: All regressors are normalized to have range [0, 1].

the power of the proposed method and of the Parzen-window estimator respond similarly to the experimental factors. The marginal effects of offset and effective magnitude are strongly negative and positive, respectively, but the offset-effective magnitude interaction effect is overwhelmingly large and positive. Coverage has a weak marginal effect on power, but it enters more strongly in the interaction with effective magnitude and in the three-way interaction. NOrMAL's errors are mostly driven by offset and its interaction with effective magnitude, as indicated by the ANOVA and logistic regression results.

Taken together, these results demonstrate that the proposed method offers better performance than standard methods in the field for estimating cluster locations from high-coverage paired-end data. The proposed method's greatest benefits are apparent when focusing on local concentrations and alternative positioning.

4.2.2. *Local Concentrations*

We next examine the ability of the proposed method to detect local concentrations in the distribution of nucleosome positions, a quantification of small-scale structure. We focus on detecting small regions of excess local concentration using the $C_{p,l}(k)$ estimand, defined in Equation (8). For this analysis, we fix $l = 73$ and $p = 3$, and use the calibrated detection procedure described in Section 3.4 with a maximum FDR of 5%.[1] For primary positions, we declare a successful detection if the best-match distance is less than 5 bp between a detected position and the true primary position. For alternative positions, we declare a successful detection if the best-match distance between a detected

[1] We reduce any contiguous sequences of detections to their mean position for interpretability. This is conservative in terms of FDR control and provides a more stringent test of the proposed methodology.

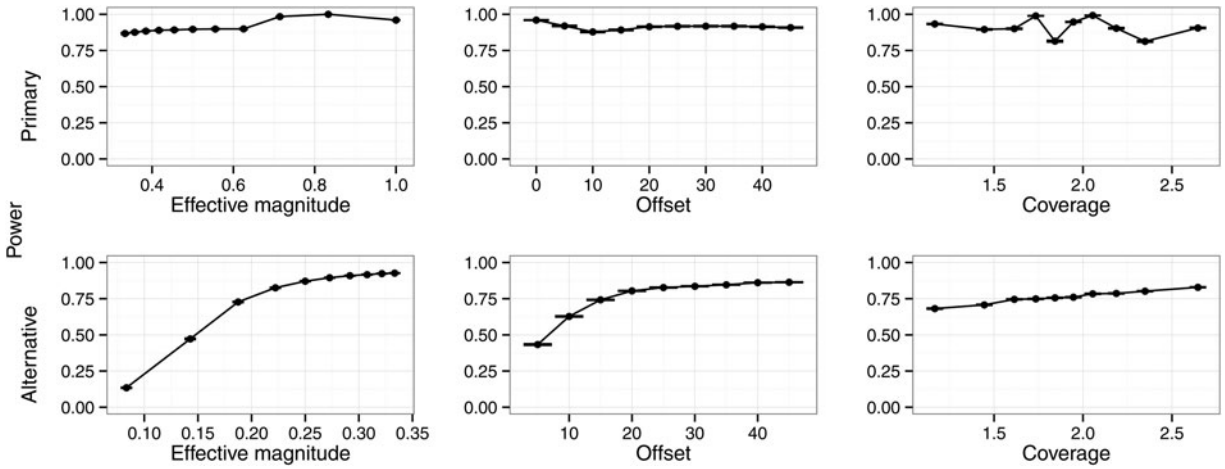

Figure 8. Power versus effective magnitude (left), alternative position offset (center), and coverage (right) for detection of primary and alternative positions ±3 bp.

position and the true alternative position is less than 1/2 of the alternative position's distance from its primary position. Figure 8 summarizes our results for primary and alternative positions, showing the power of the proposed method against the effective magnitude of the primary position, the offset of the alternative positions, and gene-level coverage. Table 3 provides the results of a design-based ANOVA of the mean absolute errors of estimated cluster locations, by gene, and Table 4 provides the results of a logistic regression of power on the design factors.

Power ranges from approximately 64% to 100% for primary positions and from approximately 2% to 100% for alternative positions across all combinations of factors, while mean absolute position errors range from 0.389 bp to 6.61 bp and from 2.17 bp to 41.8 bp, respectively. The sensitivity of power and absolute position errors to the experimental factors differs between primary and alternative positions.

For primary positions, the power increases as the effective magnitude of the primary position increases and decreases as the offset to the alternative position increases. The ANOVA results in Table 3 suggest that the majority of variation in absolute estimation errors for primary positions (75% of total and 90% of explained) is driven by the relative magnitude of primary and alternative positions. Coverage plays a minor role in the variation of these errors, even when including all of its interactions. The logistic regression results in Table 4 tell a similar story, with effective magnitude of the primary position and its interaction with the offset to the alternative position showing a strong positive effect on power. Other effects are considerably smaller.

For alternative positions, the power increases as effective magnitude of the primary position, the offset to the alternative position, and the coverage increase. The ANOVA results in Table 3 show that the majority of the variation in absolute estimation errors for alternative positions is accounted for by the offset-magnitude interaction (52% of total, 53% of explained), with the marginal contributions of magnitude, offset, and coverage accounting for most of the remaining variation (41% of total, 42% of explained).

The logistic regression results in Table 4 support these findings and shed more light on the drivers of power for primary and alternative positions. The marginal effect of effective magnitude of the primary position is similar for primary and alternative positions, but the offset-effective magnitude and three-way interactions are far stronger for alternative positions than they are for primary positions. Coverage also has a pronounced effect on power for alternative positions, both marginally and through the interaction terms.

Taken together, these results demonstrate that the proposed method can detect local concentrations in the distribution of nucleosome positions across a broad range of realistic conditions. We can reliably detect and estimate the locations' primary positions with average power over 90% and average absolute position errors of only 2.1 bp. Although alternative positions are more difficult to detect, the proposed method provides reliable inferences about their positions as well, yielding an average power of 76% and mean absolute position errors of 6.0 bp. We

Table 3. Analysis of variance of absolute position errors for the detection of primary and alternative positions using local concentration estimands.

	Primary positions			Alternative positions		
	Df	Sum Sq	*F* value	Df	Sum Sq	*F* value
Coverage	9	1453.02	63.02*	9	11136.04	1825.67*
Offset	9	5837.60	253.20*	8	34614.74	6384.17*
Magnitude	9	116496.93	5052.88*	9	74825.85	12267.12*
Coverage:Offset	81	147.78	0.71	72	5331.20	109.25*
Coverage: Magnitude	81	1292.48	6.23*	81	4532.05	82.55*
Offset:Magnitude	72	2964.07	16.07*	72	154557.47	3167.31*
Coverage:Offset: Magnitude	648	1043.81	0.63	648	3968.16	9.04*
Residuals	10090	25847.85		8100	5489.74	

NOTE: *indicates that a factor was statistically significant with $p < 0.0001$. Remaining factors had *p*-values larger than 0.95.

Table 4. Logistic regression of power on design factors as continuous variables for primary and alternative positions.

	Primary positions		Alternative positions	
	Estimate	*z* value	Estimate	*z* value
(Intercept)	1.7018	2.13	−1.6631	−44.64
Coverage	−0.0456	2.33	0.4591	7.12
Offset	0.4778	0.82	−1.5089	−20.65
Effective magnitude	2.0866	14.07	2.1086	38.37
Coverage·Offset	−0.5585	−1.42	1.0953	8.64
Coverage·Effective magnitude	−0.7448	−5.31	−0.3608	−3.75
Offset·Effective magnitude	0.9623	2.62	7.8374	55.50
Coverage·Offset·Effective magnitude	−0.3217	−0.95	8.7109	31.84

NOTE: All regressors are normalized to have range [0, 1].

discuss the implications of these capabilities for biological analyses in Section 6.

5. Real Data Analysis

Here, we illustrate the proposed methodology on real data. High-throughput sequencing data were collected on *S. cerevisiae* cell populations growing in a high-phosphate medium. The data consist of two lanes of sequencing, referred to as technical replicates, on each of two separate samples with different enzymatic digestion, referred to as biological replicates. Analyses with the proposed methods are highly reproducible, as we show in Section 5.1, and provide new insights on the fine-grained structure of nucleosome positioning. The biological relevance of these substantive findings is detailed elsewhere (Zhou et al. 2016).

In Section 5.1, we assess the reproducibility of our inferences for cluster-level summaries of nucleosome positioning (Section 5.1.1) and the locations of local concentrations (Section 5.1.2). We also compare the reproducibility of our cluster-level inferences to those obtained from Parzen-window methods and read-based estimators of the cluster-level estimands. In Section 5.2, we discuss the efficiency and scalability of inference via parallel Hamiltonian MCMC sampling. In Section 5.4 we illustrate the proposed methods on yeast and human data.

To perform inference throughout this section, we set $\mu_0 = 0$, $\tau_0 = 1/10$, $\alpha_0 = 7$, and $\gamma_0 = 10$. These values were chosen to be weakly informative on the basis of prior biological information. These values of α_0 and γ_0 imply that there is a 99% prior probability that 0.2%–13% of base pairs have β_k greater than or equal to 10 times their median. We found little sensitivity of our inferences to these choices of parameter values on a set of analyses on chromosome I. Varying τ_0 over two orders of magnitude (0.01–1) showed little effect on inferences, as did similar changes to (α_0, γ_0).

5.1. Reproducibility

We compared the reproducibility of estimates of cluster-level properties from the proposed method to those from a Parzen-window estimator, and assessed the reproducibility of estimated local concentration locations from the proposed method. We did not include NOrMAL because of its low power on simulated data. For this comparison, we used measurements from two distinct samples (biological replicates, indexed by i), each of which was sequenced twice (technical replicates, indexed by j). This design yields four datasets, H_{ij} for $i, j = 1, 2$, which allow for two comparisons within biological replicates (i.e., H_{11} vs. H_{12}, and H_{21} vs. H_{22}), and four comparisons across biological replicates. Biological replicates have different levels of enzymatic digestion, allowing us to directly assess gains in robustness from estimation of the digestion variability template, introduced in Section 2.1.

We examine the reproducibility of inferences on cluster-level and local concentration estimands in Sections 5.1.1 and 5.1.2, respectively. For these analyses, we first matched inferred positions within pairs of replicates. We then took the union of all matched positions, within each pair of replicates, as a basis for subsequent analyses; for instance, to estimate the distribution of distances between matched positions, and the correlations of inferred measures associated with each position. The same matching procedure was used for inferences on cluster-level properties and local concentrations. We present detailed results for each class of estimand below.

5.1.1. Clusters

We assessed the reproducibility of estimated cluster positions and cluster-level summaries from our method and the standard Parzen-window technique. For the former, all estimates are posterior means of the estimands specified in Equations (9)–(11) ($L_{i,j}$, $S_{i,j}$, and $R_{i,j,q}$) using a window of ±73 bp around each estimated cluster center. We set $q = 0.9$ for the sparsity estimand. For the latter, we estimated cluster-level properties using the observed read counts $\boldsymbol{y}$ directly to obtain estimates of the localization, sparsity, and structure indices described in Section 2.3 for the clusters identified by the Parzen-window method. In addition to matching inferred positions between replicates for each method, as described above, we also matched inferred positions between methods within each replicate to asses the comparability of estimates obtained by the different methods. Our results are summarized in Figure 9.

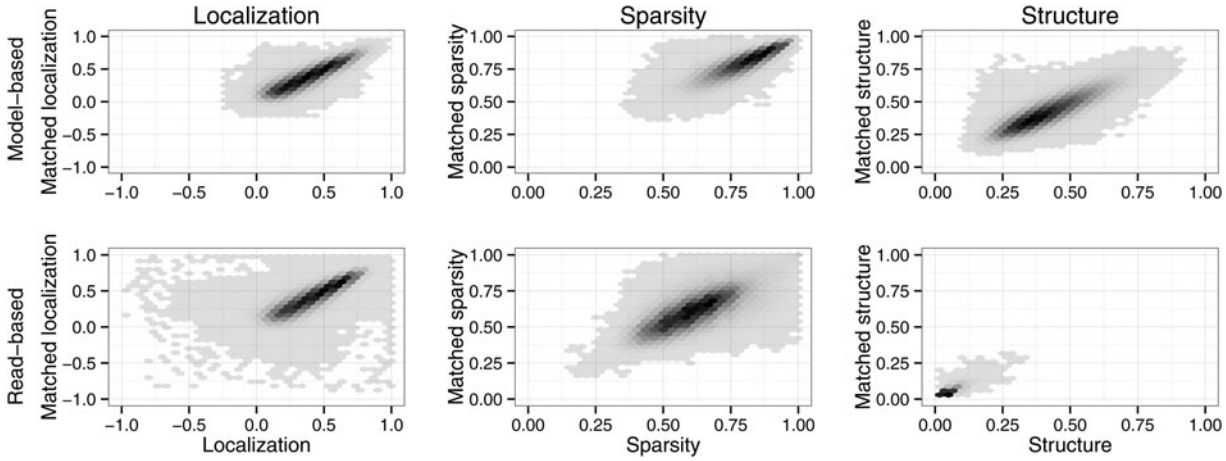

Figure 9. Joint distributions of local, structure, and sparsity indices for matched clusters between biological replicates for model-based (top) and Parzen-window/read-based (bottom).

Inferred cluster positions were highly reproducible with a mean best-match distance of 15.72 ± 0.14 bp and median best-match distance of 4 bp, between biological replicates, and of 14.30 ± 0.2 bp and 3 bp, respectively, between technical replicates. With the proposed method, 90% of clusters were matched within 44 bp across biological replicates, and within 35 bp across technical replicates. These results are comparable to the those obtained with a Parzen-window estimator, which achieves mean and median best-match distances of 15.24 ± 0.14 bp and 4 bp, between biological replicates, and of 13.98 ± 0.19 bp and 8 bp, respectively, between technical replicates. Inferred cluster positions were also consistent across methods, within each replicate, with mean and median best-match distances of 3.11 ± 0.07 bp and 1 bp. Across methods, 90% of inferred cluster positions were matched within 2 bp and 95% were matched within 3 bp.

Cluster-level properties showed significant differences between the model-based and Parzen-window estimates in reproducibility and comparability, as Figure 9 shows. The model-based estimator of the localization estimand L showed the greatest reproducibility with an R^2 of 0.765 ± 0.002 between matched clusters for biological replicates (0.799 ± 0.002 for technical replicates), performing better than the read-based estimates that had R^2's of 0.713 ± 0.003 and 0.745 ± 0.005, respectively. The model-based estimator of the structure estimand S was close behind with R^2's of 0.749 ± 0.002 and 0.795 ± 0.002 for biological and technical replicates. The read-based estimator of S fared considerably worse with R^2's of only 0.664 ± 0.003 and 0.698 ± 0.004, respectively. The model-based estimator of the sparsity estimand R showed the largest gap in reproducibility between model-based and read-based estimators, with R^2's of 0.720 ± 0.002 and 0.736 ± 0.002 for the model-based method (between biological and technical replicates) and R^2's of only 0.403 ± 0.007 and 0.526 ± 0.005 for the read-based estimator, respectively.

Localization (L) was also the most comparable feature between the model-based and read-based estimators with a Spearman correlation of 0.950 ± 0.001 within replicates. This can be seen graphically in the leftmost panels of Figure 9: the read-based localization index is noisier than the model-based one, but their distributions appear comparable otherwise. The structure index (S) was moderately comparable between the model-based and read-based estimators with a Spearman correlation of 0.784 ± 0.001. The magnitudes of these estimators are less comparable, with the model-based estimator spanning nearly three times the range of the read-based one. The sparsity index (R) was barely comparable between estimators, as one would expect from the middle panels of Figure 9. Its Spearman correlation was only 0.218 ± 0.003, and the read-based estimator spanned a far wider range of values than the model-based one. These differences arise because the model-based and read-based estimators are actually estimating different quantities. Read-based estimators are estimating properties of both the experimental errors and the distribution of positions, whereas the model-based estimators are targeting only the underlying distribution of nucleosome positions.

These results show that the proposed method provides reproducible summaries of the local arrangement of nucleosome positions despite biological and technical variation, including changes to the degree of enzymatic digestion. It outperforms standard Parzen-window and read-based estimators and provides a richer, more robust view of the underlying distribution of nucleosome positions.

5.1.2. Local Concentrations

The locations of detected local concentrations (based on $C_{3,147}(k)$) are highly reproducible across both biological and technical replicates. These results are summarized in Figure 10, where we compare the distributions of best-match distances between biological and technical replicates.

Reproducibility is higher between technical replicates than between biological replicates, with median best-match distances of 1 bp and 2 bp, respectively. Between biological replicates, 75% of positions were matched within 10 bp, 80% were matched within 15 bp, and 90% were matched within 33 bp. Between technical replicates, the corresponding quantiles were 6 bp, 11 bp, and 30 bp. These results demonstrate the reliability of the proposed method in analyzing high-throughput sequencing data, and provide confidence that the small-scale features identified by the proposed method represent real biological structure.

5.2. Parallel HMC Performance

The parallel HMC sampler performed well on real and simulated datasets, based on standard MCMC diagnostics. For the actual and simulated datasets used in Section 4, we ran 2000 iterations,

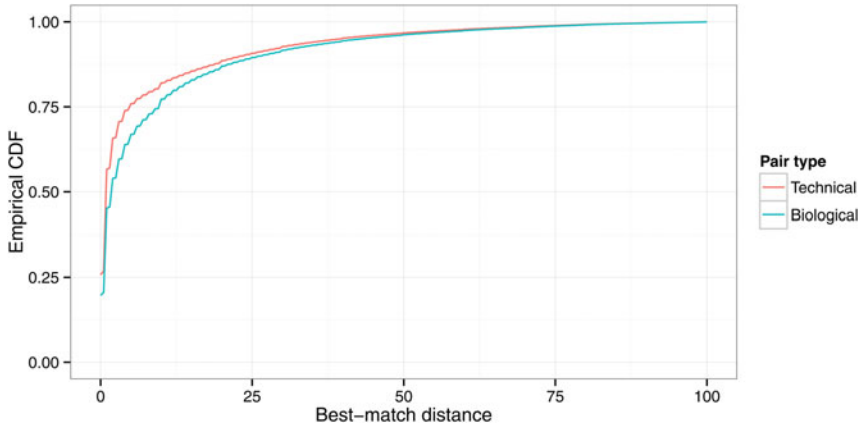

Figure 10. Empirical CDFs of best-match distances between detected local concentrations for technical (red) and biological (blue) replicates.

discarding the first 200 as burn-in. This yielded 1800 draws for $\boldsymbol{\beta}$, $\boldsymbol{\mu}$, and σ^2. The mean effective sample size for the elements of log $\boldsymbol{\beta}$ in the real dataset was 1573, with 99% of the coefficients having effective sample sizes between 304 and 2057. For the simulated dataset, the mean was 1675 with 99% between 520 and 2011. Gelman-Rubin diagnostics based on a set of MCMC runs with dispersed initializations on the smallest chromosome (I) showed multivariate potential scale reductions of 1.05 or less.

Our sampler proved sufficiently scalable for the study of *S. cerevisiae*. Using 144 cores on the Harvard Odyssey cluster and setting $B = 2000$, each simulated chromosome required 1.83 sec per iteration for a total runtime per chromosome of approximately 1 hr. The smallest *S. cerevisiae* chromosome (I) required 0.136 sec per iteration, while the largest (IV) required 0.699 sec per iteration, yielding total runtimes per chromosome of 4.5 min and 23.3 min, respectively. Running the entire *S. cerevisiae* genome required approximately 3.24 hr, using 144 cores on the Harvard Odyssey cluster. The sampler was also run on an Amazon EC2 cluster with 512 cores, processing the same genome in under an hour. In both cases, these numbers are total runtime and total number of cores; chromosomes were processed one after the other (i.e., approximately 3′45″ per chromosome on Amazon EC2), and inference on each chromosome was performed with the distributed HMC sampler.

This scales well enough for whole-genome analysis of small eukaryotic genomes such as *S. cerevisiae*, *P. falciparum*, or *C. elegans*. The runtime results also suggest that the proposed distributed inference algorithm, as implemented, is immediately useful for applications in the 10–100 MB range, which includes the entire human exome. Whole-genome analysis of larger mammalian genomes such as *M. musculus* or *H. sapiens* would require an optimized implementation of this algorithm in C/C++. With the current implementation, targeted analysis of human-scale data is possible, however. The proposed method provides the ability to *zoom in* on selected regions within large genomes, for instance using simple Parzen-window methods first, then providing high-resolution estimates of nucleosome positioning within areas believed to have regulatory or other biological significance. For example, the method could be employed only on coding sequences and their promoters, or on DNase I-hypersensitive sites (DHS) regions (Crawford et al. 2006a,b; The ENCODE Project Consortium 2011), as we illustrate in Section 5.4. Given the consistency of deconvolution- and read-based cluster location and localization estimates, computationally inexpensive read-based estimates could also be used to select regions of interest.

5.3. Evaluation on Human Data

To evaluate the performance of the proposed methods with more complex mammalian data, we analyzed the MNase midpoint data from Gaffney et al. (2012). We considered all the genes on human chromosome 20, for a total of 1721 genes, including both introns and exons. We fitted the proposed model on these data, and run detection with a 5% FDR to obtain a number of peak calls. We first looked at the distance between each cluster-level peak called by Gaffney et al. (2012) on these genes and compared them to the nearest local concentration detections based on C3147 (k). The vast majority of the peak calls (approximately 90%) had corresponding local concentration detections within 14 base pairs. Half of the peak calls (approximately 50%) had corresponding local concentration detections within 10 base pairs. The full empirical CDF of these distances is provided in Figure 11.

This analysis illustrates how the proposed method produces localized results that are consistent with published cluster-level results. Incidentally, this analysis also shows that the proposed method can be successfully applied to mammalian datasets. In Section 5.4, we go into greater depth to illustrate how the proposed method provides a finer, more detailed view of nucleosome positioning within regions of biological interest.

5.4. Biological Findings

Here, we illustrate the type of biological features template-based methods enable to identify, at a fine level of resolution, by post-processing the estimands introduced in Section 2.3, when analyzing the MNase data described above. The relevance of these findings for molecular regulation dynamics, in yeast and human, is detailed elsewhere (Zhou et al. 2016).

First, we analyzed data from an unpublished dataset of MNase titrations. For this dataset, our collaborators modulated six increasing degrees of MNase for the same biological sample before sequencing. We examined the distributions of pairwise

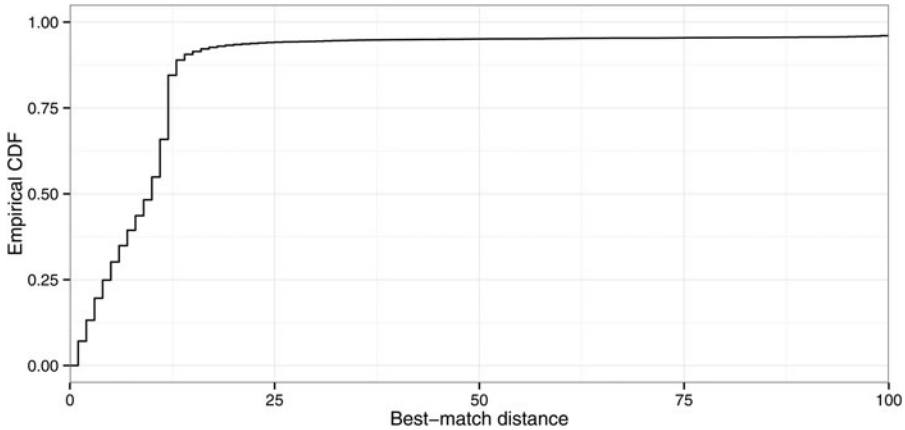

Figure 11. Empirical CDF of best-match distances between Gaffney et al. (2012) cluster-level peak calls and local concentration detections based on $C_{3,147}(k)$ at 5% FDR. The range of distances shown covers 96% of the cluster-level peak calls in Gaffney et al. (2012).

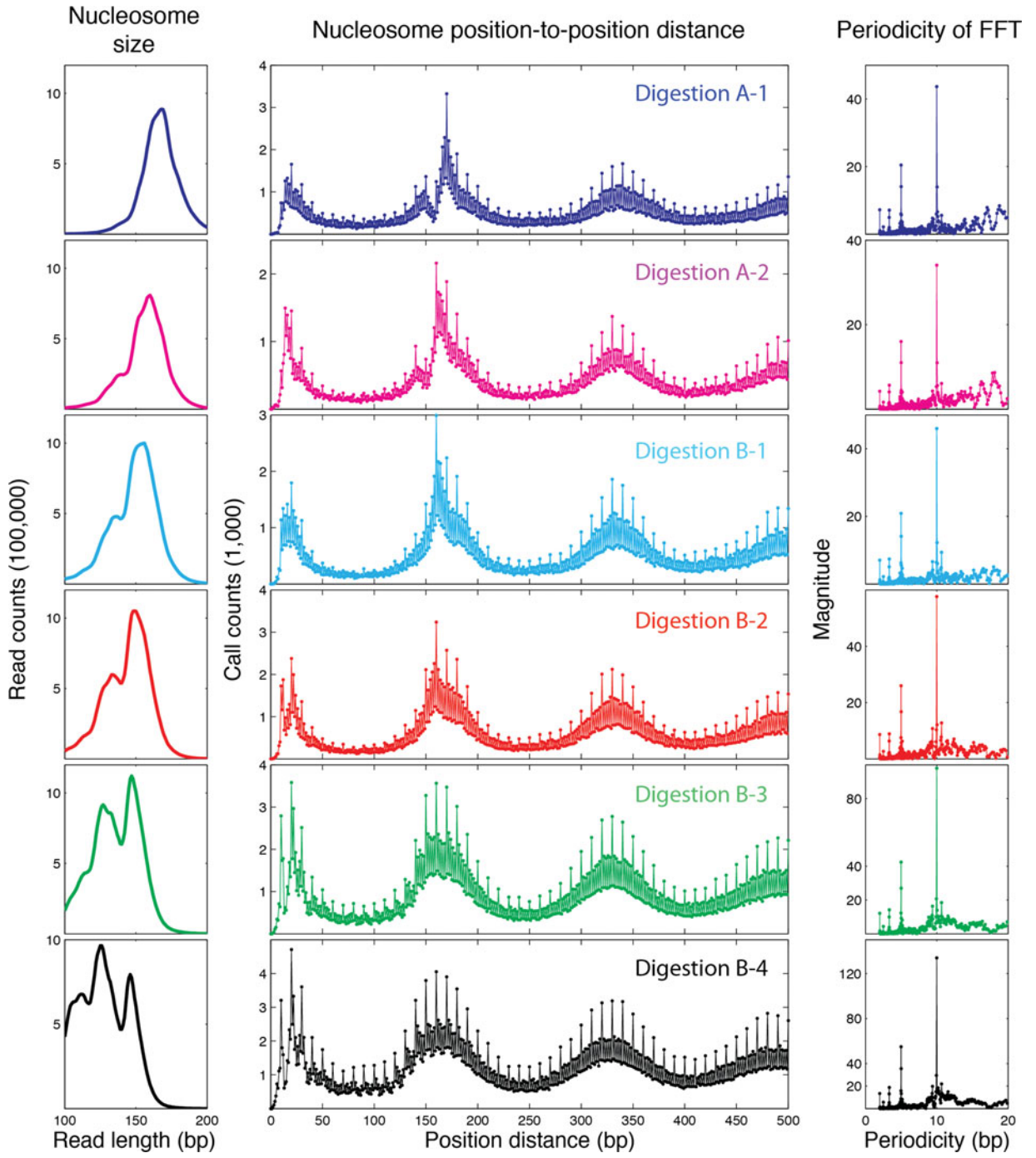

Figure 12. Robustness of inference on nucleosome-to-nucleosome positions and DNA-nucleosome binding periodicity (Brogaard et al. 2012) achieved with the template-based method to control for digestion variation in six different lanes of sequencing.

distances between detected local concentrations. Figure 12 illustrates the robustness of inference achieved with the template-based methods to control digestion variation in six different lanes of sequencing. The estimated digestion variation templates for six different lanes of sequencing are shown in the left column. The corresponding nucleosome-to-nucleosome position distances are shown in the middle column; these are estimated positions, controlling for digestion variation by leveraging the template on the left column. The corresponding FFT coefficients are shown in the right column.

Overall, the results in this Figure show the level of consistency in detecting a biological feature of interest—here, the 10 base-pair periodicity in DNA binding (Brogaard et al. 2012)—that can be expected when using a template-based approach to control digestion variation. Notably, despite the modeling assumptions do not include periodicity explicitly, the proposed methodology is quite capable of extracting periodic structure from real data, consistently, in the face of increasing variability due to digestion.

Figure 13 shows two illustrative regions of *CBFA2T2* on chromosome 20. These were selected using the two-stage strategy described in Section 5.2 to perform data analysis at scale, comparing a DHS region, as identified by Roadmap Epigenomics Consortium et al. (2015), near this gene's promoter to part of

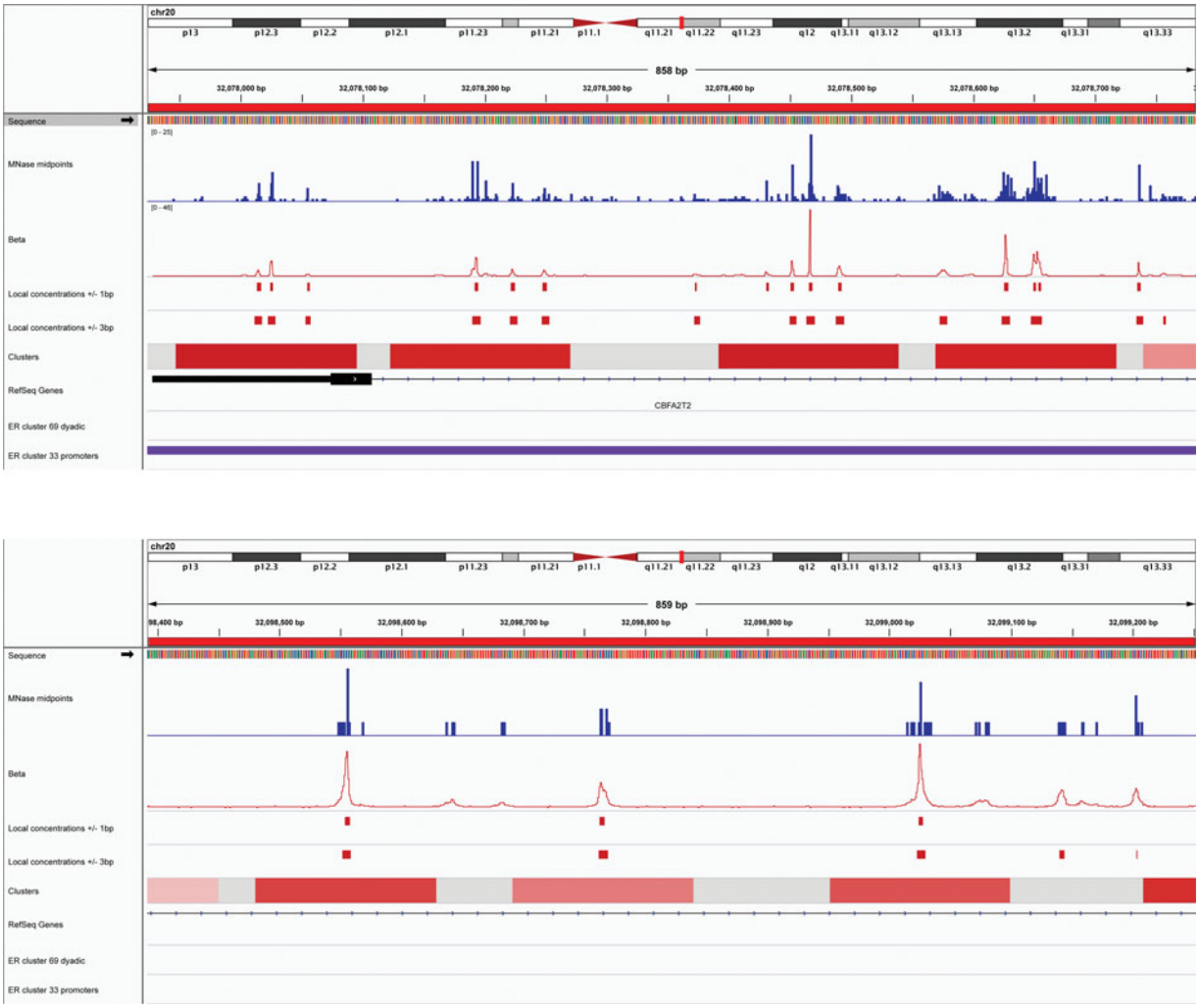

Figure 13. Illustration of alternative nucleosome position structures in two subregions of human gene *CBFA2T2*, on chromosome 20, using data by Gaffney et al. (2012). The top panel shows a DHS region with multiple detected nucleosome positions within each 147 bp-wide cluster. The bottom panel shows a non-DHS region with regular nucleosome positioning.

one of its introns. Within the DHS region, we observe ubiquitous alternative positioning by both the $C_{1,147}(k)$ and $C_{3,147}(k)$ estimands. In contrast, the intronic region shows comparatively sparse, regular nucleosome positioning, as expected. This demonstrates the use of the proposed method to zoom in on alternative positioning within selected regions of the genome, enabling finer-grained associations with local sequence features and other epigenetic features.

6. Concluding Remarks

We have presented an approach to modeling and inference about the genome-wide distribution of nucleosome positions from paired-end sequencing data. The results presented in Section 5 demonstrate the utility of the proposed methods for biological analyses, particularly reproducibility of the inferences across experimental conditions and its performance relative to other methods. Below, we expand on several broader points that have informed the development of these methods, including the lack of utility of single-cell constraints in analyses of measurements on cell populations, the relationship between estimands of interest and the performance gains stemming from model-based inferences on them, and the role of distributed computing in inference with massive, high-dimensional datasets.

6.1. Modeling

We explicitly choose not to include prior information on nucleosome spacing in this model. Previous works have used the empirically observed 150–200 bp spacing between nucleosomes within individual cells to constrain inferences on nucleosome positions (e.g., see Yuan et al. 2005; Shivaswamy et al. 2008). In the presence of alternative nucleosome positioning and chromatin dynamics, constraints on spacing that hold on a single-cell level need not hold after measurements are aggregated across a population of cells (Small et al. 2014). With sequencing coverage on the order of 10–100, only a tiny fraction of the cells in the population contribute to the observed data within each small region of the genome. The probability of observing even two reads from the same cell within, for instance, a single ORF is minuscule. As a result, single-cell constraints provide few constraints on the range of probable observations in high-throughput sequencing experiments in the presence of alternative positioning. Thus, we choose not to

use information on expected separation among nucleosomes along the sequence to constrain the inferred nucleosome positions in the proposed model. Instead, we opt for a simpler hierarchical structure within each segment, modeling locally shared distributions of nucleosome localization.

The proposed method uses information about the fragment lengths between pairs of reads that is provided by paired-end sequencing technology to infer the effects of enzymatic digestion on the measurements $\boldsymbol{y}$. Many studies have used single-end sequencing technology, which does not provide fragment lengths. The model and inference presented in Sections 2 and 3 can be adapted to this single-end context with an appropriate modification of the template $\boldsymbol{t}$ and of the digestion matrix X. Alternative sources of fragment length information, such as Bioanalyzer (e.g., Mueller 2000), could be used to estimate $\boldsymbol{t}$ in these settings. Arguably, the paired-end case will be more relevant for future biological research as its costs continue to fall and its utility expands for other applications (e.g., Katz et al. 2010; Katz et al. 2014; Katz et al. 2015).

6.2. Estimands

In defining the estimands of biological interest, we aimed to separate properties of the distribution of observed reads, which include the effects of enzymatic digestion, PCR, and sequencing, from the distribution of nucleosome positions, which is the true target of biological investigations. Existing estimators defined directly as functions of the read counts confound these distributions, impairing reproducibility of the analysis and ultimately their utility for scientific exploration. In the model introduced in Section 2, the distribution of nucleosome positions corresponds to the $\boldsymbol{\beta}$ vector, while the template $\boldsymbol{t}$ and the remaining error structure capture other sources of experimental variation. The estimands introduced in Section 2.3 are functions only of the true underlying $\boldsymbol{\beta}$, and are thus unaffected by variation due to the experimental process, at least in principle. Below, we discuss two subtle points on the construction of these estimands.

First, there is a key distinction between the cluster-based estimands, like $L_{i,j}$, $S_{i,j}$, and $R_{i,j,q}$, and summaries of local structure, like those based on $C_{p,l}(k)$. Cluster-based estimands capture properties of the distribution of nucleosome positions within small regions identified by a clustering algorithm. Thus, these measures depend on the particular definition of "cluster" and on the clustering method used. The sensitivity of these estimands to these choices is often substantial. Fine-grained structure is lost in the reduction of data to clusters. These problems may be balanced by possible gains in interpretability, due to the much more concise cluster structure, and by the comparisons of large-scale nucleosome structures these measures enable. In contrast, local estimands such as the local concentration index $C_{p,l}(k)$ summarize properties of the distribution of nucleosome positions locally, without relying on a clustering criterion. They lead to reproducible analyses and can be relied upon for scientific discovery of small-scale features. These local estimands are most useful in combination with the cluster-level estimands, since they provide a complementary view on the distribution of nucleosome positions.

Second, we have found that the magnitude of the performance gains stemming from model-based inferences depends on the estimand of interest, and on the criterion for assessing performance. For instance, the proposed method outperforms read-based estimators in terms of power and error when targeting the cluster-level localization estimand, L, but the difference in reproducibility is not overwhelming. On the other hand, the increase in reproducibility one can expect from the proposed method is substantial when targeting structure and sparsity measures, S and R. This result reflects the greater sensitivity of read-based estimators of structure and sparsity to noise in the observations. In addition, as we have shown in Sections 4.2.2 and 5.1.2, the proposed method can provide reproducible inferences about local features less than 10 bp wide, whereas inference on properties of the distribution of nucleosome positions at such a fine resolution has been so far considered infeasible MNase-seq data. More generally, the more sensitive an estimand is to noise in the observations, the greater the performance gains expected from using the proposed method are.

Overall, our results suggest that careful probabilistic modeling of the core sources of experimental variation can enable scientific inferences at a previously infeasible scale.

6.3. Inference

The use of distributed computing was essential for fitting the proposed model, as it allowed us to sample from the marginal posterior of $\boldsymbol{\beta}$ in only minutes per chromosome. We leveraged the conditional independence structure encoded by our model to create an efficient, scalable, distributed MCMC sampler. This structure stems from the finite length of the digestion variability template $\boldsymbol{t}$. As the template is $2w + 1$ wide, subvectors of $\boldsymbol{\beta}$ separated by at least $2w$ base pairs are conditionally independent a posteriori given $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$. Thus, we can update collections of such subvectors independently across hundreds of processors. The communication costs involved in this procedure are low, as only each subvector (padded by w entries on each end) and the relevant entries of $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ are needed for each update.

To update each subvector of $\boldsymbol{\beta}$ in the MCMC sampler, we use a simple HMC step. Because of the convolution structure of X, computation of the conditional posterior and its gradient for B-entries-long subvectors of $\boldsymbol{\beta}$ scale as $O(B \log B)$. For a fixed block size B, adding processors in proportion to the length of the chromosome being analyzed maintains constant runtime. In addition, the proposed method has constant runtime with respect to the number of fragments observed, omitting alignment. As shown in Section 5.2, this scalable inference strategy leads to a high-quality sampler.

We propose a combination of Bayesian and frequentist techniques for the detection of local concentrations of nucleosomes. We use the local concentration estimands, $C_{p,l}(k)$, to target structures of interest. We then use draws from the parallel HMC sampler to estimate the posterior probabilities that these estimands exceed their expected value under a locally uniform distribution of nucleosome positions. Instead of using these estimated posterior probabilities to make inferences directly, we calibrate them using frequentist multiple testing techniques (Storey and Tibshirani 2003). And instead of relying upon the model to provide a null distribution, we adopt a data-dependent

permutation null in the spirit of Fisher's exact test. The calibration step provides guarantees on the behavior of the detection procedure under a permutation null and transforms Bayesian posterior probabilities to the more standard and interpretable scales of FDRs and q-values.

The pragmatic approach to detection described above is one of several steps in our overall approach to the analysis of nucleosome positioning. First, we use a probability model to build statistics that directly target the scientific estimands of interest, and perform inference with MCMC. Second, we use permutations to define a reference distribution based on the observed data and the segmentation, and detect local concentrations of nucleosome positions. Third, we evaluate the power, accuracy, and reproducibility of inferences from our method using biologically motivated simulations, and technical and biological replicates. Each step in this overall strategy reflects less reliance upon modeling assumptions and a greater emphasis on external validity. The success of this strategy is reflected in the empirical results and simulation studies presented in Sections 4 and 5. We obtain accurate, reproducible, scalable inferences about the genome-wide distribution of nucleosome positions with well-studied operating characteristics, providing new capabilities to this area of biology.

Throughout this work, we have focused on the comparison of replicates. In practice, it is often of interest to combine information from multiple replicates to obtain more sensitive, secure inferences. The method presented can be adapted in one of several ways. If digestion variation is consistent among replicates, the reads can be combined into a common set for template estimation and inference, enabling the use of the given method without modification. If digestion variation differs among replicates, the given model can be extended to allow for separate observations with separate digestion templates. Applying the model separately to each replicate and merging the results is simpler and likely more robust to unmodeled source biological variation. These advantages come at the cost of some power to detect marginal positions; however, we believe that it is more likely to yield robust nucleosome positions of biological interest. Methods such as the irreproducible discovery rate of Li et al. (2011) are well suited for the combination of such results from multiple replicates.

6.4. Biological Significance

In follow-up work, we show that the high-resolution positions that can be reliably estimated with the methods proposed here reveal novel aspects of chromatin organization, such as alternatively positioned nucleosomes and periodic occurrences of dinucleotide sequences relative to nucleosome dyads, in vivo, in both yeast and human cells (Zhou et al. 2016). Our results suggest that alternatively positioned nucleosomes at transcription start sites represent different states of promoter nucleosomes during transcription initiation—one state in which a nucleosome competes with the transcription machinery and another that permits transcription preinitiation complex assembly. The analysis presented in this work further indicates that the proposed method can be applied to precisely map the positions of nucleosomes in diverse organisms, revealing novel aspects of chromatin function.

Supplementary Materials

The supplement provides details of the distributed HMC algorithm, and details an approximate EM algorithm that can be used to provide starting values for the HMC sampler. It also includes additional Figures for the power analysis presented in Section 4.2.

Acknowledgments

The authors thank Xu Zhou and Erin O'Shea for providing data and valuable insights, the FAS Science Division Research Computing Group at Harvard University, and the editor and anonymous reviewers for their valuable input.

Funding

This work was partially supported by the National Science Foundation under grants IIS-1017967, DMS-1106980, and CAREER IIS-1149662, and by the National Institute of Health under grant R01 GM096193.

References

Albert, I., Mavrich, T. N., Tomsho, L. P., Qi, J., Zanton, S. J., Schuster, S. C., and Pugh, B. F. (2007), "Translational and Rotational Settings of H2A.Z Nucleosomes Across the Saccharomyces Cerevisiae Genome," *Nature*, 446, 572–576. [967,976,977]

Barski, A., and Zhao, K. (2009), "Genomic Location Analysis by ChIP-Seq," *Journal of Cellular Biochemistry*, 107, 11–18. [968]

Brogaard, K., Xi, L., Wang, J.-P., and Widom, J. (2012), "A Map of Nucleosome Positions in Yeast at Base-Pair Resolution," *Nature*, 486, 496–501. [968,983]

Cairns, J., Spyrou, C., Stark, R., Smith, M. L., Lynch, A. G., and Tavaré, S. (2011), "BayesPeak–an R Package for Analysing ChIP-seq Data," *Bioinformatics*, 27, 713–714. [968]

Crawford, G. E., Davis, S., Scacheri, P. C., Renaud, G., Halawi, M. J., Erdos, M. R., Green, R., Meltzer, P. S., Wolfsberg, T. G., and Collins, F. S. (2006a), "Dnase-chip: A High-resolution Method to Identify Dnase i Hypersensitive Sites Using Tiled Microarrays," *Nature Methods*, 3, 503–509. [982]

Crawford, G. E., Holt, I. E., Whittle, J., Webb, B. D., Tai, D., Davis, S., Margulies, E. H., Chen, Y. D., Bernat, J. A., Ginsburg, D., Zhou, D., Luo, S., Vasicek, T. J., Daly, M. J., Wolfsberg, T. G., and Collins, F. S. (2006b), "Genome-Wide Mapping of Dnase Hypersensitive Sites Using Massively Parallel Signature Sequencing (mpss)," *Genome Research*, 16, 123–131. [982]

Flores, O., and Orozco, M. (2011), "nucleR: A Package for Non-Parametric Nucleosome Positioning," *Bioinformatics*, 27, 2149–2150. [968]

Frazee, A. C., Sabunciyan, S., Hansen, K. D., Irizarry, R. A., and Leek, J. T. (2014), "Differential Expression Analysis of Rna-seq Data at Single-Base Resolution," *Biostatistics*, 15, 413–426. [968,974,975]

Fu, K., Tang, Q., Feng, J., Liu, X. S., and Zhang, Y. (2012), "DiNuP: A Systematic Approach to Identify Regions of Differential Nucleosome Positioning," *Bioinformatics*, 28, 1965–1971. [968]

Gaffney, D. J., McVicker, G., Pai, A. A., Fondufe-Mittendorf, Y. N., Lewellen, N., Michelini, K., Widom, J., Gilad, Y., and Pritchard, J. K. (2012), "Controls of Nucleosome Positioning in the Human Genome," *PLoS Genetics*, 8, e1003036, 11. [967,971,982]

Gkikopoulos, T., Schofield, P., Singh, V., Pinskaya, M., Mellor, J., Smolle, M., Workman, J. L., Barton, G. J., and Owen-Hughes, T. (2011), "A Role for Snf2-Related Nucleosome-Spacing Enzymes in Genome-Wide Nucleosome Organization," *Science*, 333, 1758–1760. [968]

Gupta, M. (2007), "Generalized Hierarchical Markov Models for the Discovery of Length-Constrained Sequence Features From Genome Tiling Arrays," *Biometrics*, 63, 797–805. [968]

Jansen, A., and Verstrepen, K. J. (2011), "Nucleosome Positioning in Saccharomyces cerevisiae," *Microbiology and Molecular Biology Reviews: MMBR*, 75, 301–320. [968]

Katz, Y., Wang, E. T., Airoldi, E. M., and Burge, C. B. (2010), "Analysis and Design of RNA Sequencing Experiments for Identifying Isoform Regulation," *Nature Methods*, 7, 1009–1015. [985]

Katz, Y., Li, F., Lambert, N. J., Sokol, E. S., Tam, W.-L., Cheng, A. W., Airoldi, E. M., Lengner, C. J., Gupta, P. B., Yu, Z., Jaenisch, R., Burge, C. B. (2014), "Musashi Proteins are Post-Transcriptional Regulators of the Epithelial-Luminal Cell State," *eLife*, 3, e03915. [985]

Katz, Y., Wang, E. T., Silterra, J., Schwartz, S., Wong, B., Thorvaldsdóttir, H., Robinson, J. T., Mesirov, J. P., Airoldi, E. M., Burge, C. B. (2015), "Quantitative Visualization of Alternative Exon Expression from RNA-Seq Data," *Bioinformatics*, 31, 2400–2402. [985]

Langmead, B., Trapnell, C., Pop, M., and Salzberg, S. L. (2009), "Ultrafast and Memory-Efficient Alignment of Short DNA Sequences to the Human Genome," *Genome Biology*, 10, R25. [967]

Lee, W., Tillo, D., Bray, N., Morse, R. H., Davis, R. W., Hughes, T. R., and Nislow, C. (2007), "A High-Resolution Atlas of Nucleosome Occupancy in Yeast," *Nature Genetics*, 39, 1235–1244. [967]

Li, Q., Brown, J. B., Huang, H., and Bickel, P. J. (2011), "Measuring Reproducibility of High-Throughput Experiments," *The Annals of Applied Statistics*, 5, 1752–1779. [986]

Mitra, R., and Gupta, M. (2011), "A Continuous-Index Bayesian Hidden Markov Model for Prediction of Nucleosome Positioning in Genomic DNA," *Biostatistics*, 12, 462–477. [968]

Mo, Q. (2012), "A Fully Bayesian Hidden Ising Model for Chip-seq Data Analysis," *Biostatistics*, 13, 113–128. [968]

Mueller, O. (2000), "High Precision Restriction Fragment Sizing With the Agilent 2100 Bioanalyzer: Application Note," available at *http://www.chem-agilent.com/pdf/5968-7501EN.pdf*. [985]

Newton, M. A., Noueiry, A., Sarkar, D., and Ahlquist, P. (2004), "Detecting Differential Gene Expression With a Semiparametric Hierarchical Mixture Method," *Biostatistics*, 5, 155–176. [975]

Park, P. J. (2009), "ChIP-Seq: Advantages and Challenges of a Maturing Technology," *Nature Reviews Genetics*, 10, 669–680. [968]

Pepke, S., Wold, B., and Mortazavi, A. (2009), "Computation for ChIP-seq and RNA-seq Studies," *Nature Methods*, 6, S22–S32. [968]

Polishko, A., Lonardi, S., Ponts, N., and Le Roch, K. G. (2014), "PuFFIN: A Parameter-Free Method to Build Nucleosome Maps From Paired-End Reads," in *Proceedings of the Fourth Annual RECOMB Satellite Workshop on Massively Parallel Sequencing (RECOMB-seq)*, Pittsburgh, PA. [968]

Polishko, A., Ponts, N., Le Roch, K. G., and Lonardi, S. (2012), "NORMAL: Accurate Nucleosome Positioning Using a Modified Gaussian Mixture Model," *Bioinformatics*, 28, 242–249. [968,977]

Rashid, N. U., Giresi, P. G., Ibrahim, J. G., Sun, W., and Lieb, J. D. (2011), "ZINBA Integrates Local Covariates With DNA-seq Data to Identify Broad and Narrow Regions of Enrichment, Even Within Amplified Genomic Regions," *Genome Biology*, 12, R67. [968]

Roadmap Epigenomics Consortium, Kundaje, A., Meuleman, W., Ernst, J., Bilenky, M., Yen, A., Heravi-Moussavi, A., Kheradpour, P., Zhang, Z., Wang, J., Ziller, M. J., Amin, V., Whitaker, J. W., Schultz, M. D., Ward, L. D., Sarkar, A., Quon, G., Sandstrom, R. S., Eaton, M. L., Wu, Y.-C., Pfenning, A. R., Wang, X., Claussnitzer, M., Liu, Y., Coarfa, C., Harris, R. A., Shoresh, N., Epstein, C. B., Gjoneska, E., Leung, D., Xie, W., Hawkins, R. D., Lister, R., Hong, C., Gascard, P., Mungall, A. J., Moore, R., Chuah, E., Tam, A., Canfield, T. K., Hansen, R. S., Kaul, R., Sabo, P. J., Bansal, M. S., Carles, A., Dixon, J. R., Farh, K.-H., Feizi, S., Karlic, R., Kim, A.-R., Kulkarni, A., Li, D., Lowdon, R., Elliott, G. N., Mercer, T. R., Neph, S. J., Onuchic, V., Polak, P., Rajagopal, N., Ray, P., Sallari, R. C., and Siebe, K. T. (2015), "Integrative Analysis of 111 Reference Human Epigenomes," *Nature*, 518, 317–330. [983]

Schöpflin, R., Teif, V. B., Müller, O., Weinberg, C., Rippe, K., and Wedemann, G. (2013), "Modeling Nucleosome Position Distributions From Experimental Nucleosome Positioning Maps," *Bioinformatics*, 29, 2380–2386. [968]

Schwartzman, A., Jaffe, A., Gavrilov, Y., and Meyer, C. A. (2013), "Multiple Testing of Local Maxima for Detection of Peaks in ChIP-Seq Data," *The Annals of Applied Statistics*, 7, 471–494. [968]

Segal, E., Fondufe-Mittendorf, Y., Chen, L., Thastrom, A., Field, Y., Moore, I. K., Wang, J. Z., and Widom, J. (2006), "A Genomic Code for Nucleosome Positioning," *Nature*, 42, 772–778. [967]

Shivaswamy, S., Bhinge, A., Zhao, Y., Jones, S., Hirst, M., and Iyer, V. R. (2008), "Dynamic Remodeling of Individual Nucleosomes Across a Eukaryotic Genome in Response to Transcriptional Perturbation," *PLoS Biology*, 6, e65. [968,972,976,984]

Small, E. C., Xi, L., Wang, J.-P., Widom, J., and Licht, J. D. (2014), "Single-Cell Nucleosome Mapping Reveals the Molecular Basis of Gene Expression Heterogeneity," *Proceedings of the National Academy of Sciences*, 111, E2462–E2471. [984]

Storey, J. D., and Tibshirani, R. (2003), "Statistical Significance for Genomewide Studies," *Proceedings of the National Academy of Sciences of the United States of America*, 100, 9440. [975,985]

Sun, W., Buck, M. J., Patel, M., and Davis, I. J. (2009a), "Improved ChIP-chip Analysis by a Mixture Model Approach," *BMC Bioinformatics*, 10, 173. [968]

Sun, W., Xie, W., Xu, F., Grunstein, M., and Li, K. (2009b), "Dissecting Nucleosome Free Regions by a Segmental Semi-Markov Model," *PloS One*, 4, e4721. [968]

The ENCODE Project Consortium. (2011), "A User's Guide to the Encyclopedia of DNA Elements (ENCODE)," *PLoS Biol*, 9, e1001046. [982]

Tirosh, I. (2012), "Computational Analysis of Nucleosome Positioning," *Methods in Molecular Biology*, 833, 443–449. [967,976,977]

Tsankov, A. M., Thompson, D. A., Socha, A., Regev, A., and Rando, O. J. (2010), "The Role of Nucleosome Positioning in the Evolution of Gene Regulation," *PLoS Biology*, 8, e1000414. [967,976,977]

Valouev, A., Ichikawa, J., Tonthat, T., Stuart, J., Ranade, S., Peckham, H., Zeng, K., Malek, J. A., Costa, G., McKernan, K., Sidow, A., Fire, A., and Johnson, S. M. (2008), "A High-Resolution, Nucleosome Position Map of C. elegans Reveals a Lack of Universal Sequence-Dictated Positioning," *Genome Research*, 18, 1051–1063. [971,977]

Weiner, A., Hughes, A., Yassour, M., Rando, O. J., and Friedman, N. (2010), "High-Resolution Nucleosome Mapping Reveals Transcription-Dependent Promoter Packaging," *Genome Research*, 20, 90–100. [968,977]

Yassour, M., Kaplan, T., Jaimovich, A., and Friedman, N. (2008), "Nucleosome Positioning From Tiling Microarray Data," *Bioinformatics*, 24, i139–i146. [968]

Yuan, G.-C., and Liu, J. S. (2008), "Genomic Sequence is Highly Predictive of Local Nucleosome Depletion," *PLoS Computational Biology*, 4, e13. [968]

Yuan, G.-C., Liu, Y.-J., Dion, M. F., Slack, M. D., Wu, L. F., Altschuler, S. J., and Rando, O. J. (2005), "Genome-Scale Identification of Nucleosome Positions in *S. cerevisiae*," *Science*, 309, 626–630. [967,984]

Zhang, X., Robertson, G., Woo, S., Hoffman, B. G., and Gottardo, R. (2012), "Probabilistic Inference for Nucleosome Positioning With MNase-Based or Sonicated Short-Read Data," *PloS One*, 7, e32095. [968]

Zhang, Y., Liu, T., Meyer, C. A., Eeckhote, J., Johnson, D. S., Bernstein, B. E., Nusbaum, C., Myers, R. M., Brown, M., Li, W., and Liu, X. S. (2008), "Model-Based Analysis of ChIP-Seq (MACS)," *Genome Biology*, 9, R137. [968]

Zhou, X., Blocker, A. W., Airoldi, E. M., and O'Shea, E. K. (2016), "A Computational Approach to Map Nucleosome Positions and Alternative Chromatin States with Base Pair Resolution," *eLife*, in press. [980,982,986]

Zhu, C., Byrd, R. H., Lu, P., and Nocedal, J. (1997), "Algorithm 778: L-bfgs-b: Fortran Subroutines for Large-Scale Bound-Constrained Optimization," *ACM Transactions on Mathematical Software*, 23, 550–560. [973]

Online supplement for "Template-based models for genome-wide nucleosome analysis at base-pair resolution"

Abstract

This supplement provides details of the distributed HMC algorithm used in the paper and an approximate EM algorithm that can be used to provide starting values for the HMC sampler. It also includes additional figures for the power analysis presented in Section 4.2 of the paper.

Contents

A Algorithmic details of inference

A.1 Distributed HMC sampler

Recall that the model specified in Section 2 is:

$$y_k|\lambda_k \sim \text{Poisson } (\lambda_k) \tag{1}$$

$$\lambda_{(N\times 1)} \equiv X_{(N\times(N-\ell_0))}\ \beta_{((N-\ell_0)\times 1)}, \tag{2}$$

$$\beta_k > 0 \text{ for } k = \lfloor \ell_0/2 \rfloor + 1 \ldots N - \lfloor \ell_0/2 \rfloor$$

$$\log \beta_k \sim \text{Normal } (\mu_{s_k}, \sigma^2_{s_k}) \tag{3}$$

given a segmentation function $s : \{1 \ldots N\} \to \{1 \ldots S\}$, which maps the N base pair locations to S regions in which coefficients β_k can be assumed to be identically distributed. X specifies the contribution of a nucleosome positioned at base pair k to the expected number of reads at base pair m due to digestion variability, and $s(k)$ is denoted as s_k for compactness. This specification is completed with independent priors on each (μ_s, σ^2_s):

$$\sigma^2_s \sim \text{InvGamma}(\alpha_0, \gamma_0)\ , \tag{4}$$

$$\mu_s|\sigma^2_s \sim N(\mu_0, \frac{\sigma^2_s}{n_s \tau_0})\ , \tag{5}$$

where n_s is the length of segment s.

Our MCMC sampler alternates between two conditional updates:

1. Draw $(\boldsymbol{\mu}^{(r)}, \boldsymbol{\sigma}^{2\,(r)})|\boldsymbol{\beta}^{(r-1)}$ directly, then
2. Update $\boldsymbol{\beta}^{(r)}|(\boldsymbol{\mu}^{(r)}, \boldsymbol{\sigma}^{2\,(r)})$ via a distributed HMC step.

The former is a standard conjugate draw, while the latter is done via a distributed version of the standard Hamiltonian Monte Carlo (HMC) routine.

A.1.1 Hyperparameter updates

In detail, step 1 consists of the following draws for each (μ_s, σ_s^2), defining $\boldsymbol{\theta} = \log \boldsymbol{\beta}$:

$$\sigma_s^{2\,(r)} | \boldsymbol{\beta}^{(r-1)} \sim InvGamma\left(\frac{n_s}{2} + \alpha_0,\ \frac{1}{2} \sum_{k:s_k=s} (\theta_k^{(r-1)} - \bar{\theta}_s^{(r-1)})^2 \right. \tag{6}$$
$$\left. + \frac{\tau_0 n_s}{2(1+\tau_0)} (\bar{\theta}_s^{(r-1)} - \mu_0)^2 + \gamma_0 \right) ,$$
$$\mu_s^{(r)} | \sigma_s^{2\,(r)}, \boldsymbol{\beta}^{(r-1)} \sim N\left(\frac{\bar{\theta}_s^{(r-1)} + \tau_0 \mu_0}{1+\tau_0}, \frac{\sigma_s^{2\,(r)}}{n_s(1+\tau_0)} \right) , \tag{7}$$

where $\bar{\theta}_s^{(r-1)} = \frac{1}{n_s} \sum_{k:s_k=s} \theta_k^{(r-1)}$. These are standard conjugate updates and have computational and memory complexity $O(N)$.

A.1.2 Distributed HMC update for $\boldsymbol{\beta}$

The draws in step 2 proceed in two stages, using two partitions of $\boldsymbol{\beta}$. The first starts at the beginning of $\boldsymbol{\beta}$ and proceeds forward with subvectors of length at most B separated by $2w$, yielding

$$D_1 = \boldsymbol{\beta}_{[1:B]}, \boldsymbol{\beta}_{[B+2w+1:2B+2w]}, \ldots, \boldsymbol{\beta}_{[n_b(B+2w)+1:N]} , \tag{8}$$
$$D_2 = \boldsymbol{\beta}_{[B/2+1:3B/2]}, \boldsymbol{\beta}_{[3B/2+2w+1:5B/2+2w]}, \ldots, \boldsymbol{\beta}_{[n_b(B+2w)B/2+1:N]} . \tag{9}$$

The subvectors within each partition are conditionally-independent given the $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ and the entries of $\boldsymbol{\beta}$ separating them. Hence, we can update them in parallel across multiple processors. The basic structure of these updates follows Algorithm 1.

The individual, worker-level HMC updates are done on $\boldsymbol{\theta} = \log \boldsymbol{\beta}$ and follow the standard leapfrog-based HMC procedure outlined in Neal (2010). To compute these HMC updates, we require the log-posterior density of each subvector of $\boldsymbol{\theta}$ and its gradient. First, define $\boldsymbol{\lambda} = X\boldsymbol{\beta}$, $\boldsymbol{m} = (\mu_{s_k} \text{ for } k = 1, \ldots, N)^\top$, and $\boldsymbol{v} = (\sigma_{s_k}^2 \text{ for } k = 1, \ldots, N)^\top$. Also, for vectors

Distributed HMC update

`/* Send conditioning information */`
Broadcast $\boldsymbol{\mu}^{(t-1)}$, and $\boldsymbol{\sigma}^{2\,(t-1)}$ to all workers ;
for *offset* **in** *(0, B/2)***:**
    `/* Send first round of jobs to workers */`
    for w **in** `range(`*min(nWorkers, nBlocks)*`)`**:**
        $start = \max(0, (w-1)(B+2w) - 2w + \text{offset}) + 1$;
        $end = \min(N, w(B+2w) + 2w + \text{offset})$;
        Send $\boldsymbol{\theta}[start : end]$ to worker process w with work tag attached ;
    if *len(startVec)* $<$ *nWorkers***:**
        Pause remaining workers ;
    `/* Collect results */`
    Set *nComplete* $= 0$, *nStarted* $= \min$(nWorkers, nBlocks)
    while *nComplete* $<$ *nBlocks***:**
        Receive result $\boldsymbol{\theta}[start : end]$ from arbitrary worker with tag b_1 ;
        Incorporate result into working copy of $\boldsymbol{\theta}^{(t)}$;
        nComplete++ ;
        if *nStarted* $<$ *nBlocks***:**
            `/* Send additional jobs as needed */`
            $w = nStarted + 1$;
            $start = \max(0, (w-1)(B+2w) - 2w + \text{offset}) + 1$;
            $end = \min(N, w(B+2w) + 2w + \text{offset})$;
            Send $\boldsymbol{\theta}[start : end]$ to last completed worker process with work tag attached;
            nStarted++;

Algorithm 1: Distributed HMC update

of equal dimension, let / denote entrywise division and ** denote entrywise powers. Then,

$$\log p(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\sigma}^2) = -\mathbf{1}^\top X\boldsymbol{\beta} + \sum_k y_k \log\left(\boldsymbol{x}_k^T\boldsymbol{\beta}\right) - \frac{1}{2}\sum_k \frac{(\theta_k - \mu_{s_k})^2}{\sigma_{s_k}^2} + \text{const} \ , \tag{10}$$

$$\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\sigma}^2) = -\text{diag}\left(\boldsymbol{\beta}\right) X^\top(\mathbf{1} - \boldsymbol{y}/\boldsymbol{\lambda}) - (\boldsymbol{\theta} - \boldsymbol{m})/\boldsymbol{v} \ , \tag{11}$$

$$\nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}^\top} \log p(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\sigma}^2) = -\text{diag}\left(\boldsymbol{\beta}\right) X'WX\,\text{diag}\left(\boldsymbol{\beta}\right) - \text{diag}\left(\boldsymbol{\beta}\right) X'(\mathbf{1} - \boldsymbol{y}/\boldsymbol{\lambda}) - \text{diag}\left(\mathbf{1}/\boldsymbol{\sigma}^2\right) \ , \tag{12}$$

where $W = \text{diag}\left(\boldsymbol{y}/\boldsymbol{\lambda}\text{**}2\right)$. Due to the convolution structure of X, all matrix-vector products involving X and $X^\top$ can be reduced to convolutions of vectors with the template vector $\boldsymbol{t}$. This also enables efficient computation of the Hessian's diagonal, as

$$\boldsymbol{\lambda} = X\boldsymbol{\beta} = \left(\mathbf{0}_{\lfloor \ell_0/2 \rfloor}^\top \ \boldsymbol{\beta}^\top \ \mathbf{0}_{\lfloor \ell_0/2 \rfloor}^\top\right)^\top * \boldsymbol{t} \tag{13}$$

$$X^\top(\mathbf{1} - y/\boldsymbol{\lambda}) = (\mathbf{1} - y/\boldsymbol{\lambda}) * \boldsymbol{t} \ , \tag{14}$$

$$\text{diag}\left(X'WX\right) = (\boldsymbol{y}/\boldsymbol{\lambda}\text{**}2) * \boldsymbol{t}\text{**}2 \ . \tag{15}$$

This reduces the computational complexity of these evaluations to $O(B \log B)$ for each update of each subvector of $\boldsymbol{\beta}$. Our block-level HMC steps are detailed in Algorithm 2. In practice, we fix $\epsilon_{\min} = 0.001$, $\epsilon_{\max} = 0.1$, and $L = 100$. We also use a fixed diagonal mass matrix, although the algorithm can accommodate estimating it at every iteration if needed. However, to maintain the $O(B \log B)$ scaling of our algorithm's complexity with block size, M must remain diagonal. If non-diagonal M is used and/or estimated, our HMC update instead scales as $O(B^2)$. In either case, the overall algorithm scales $O(N)$ for given a fixed block size B. Memory requirements are $O(N)$ for the master process running the hyperparameter draws and coordinating the distributed HMC updates. Each worker process requires $O(B \log B)$ memory to run the distributed HMC updates for diagonal M, while using a non-diagonal mass matrix M requires $O(B^2)$ memory per worker.

A.2 Approximate EM algorithm

We develop an approximate EM algorithm, based on a Gaussian approximation of the conditional posterior of $\boldsymbol{\theta}$, as to obtain starting values for the MCMC sampler given in A.1. It provides a high-quality initialization for $\boldsymbol{\beta}$, $\boldsymbol{\mu}$, and $\boldsymbol{\sigma}^2$. Simpler initializations are possible, but obtaining high-quality initial estimates can greatly reduce the number of MCMC iterations required for reliable inferences.

Data: Trajectory length L, $\boldsymbol{\mu}$, $\boldsymbol{\sigma}^2$, $\boldsymbol{\theta}[start : end]$, ϵmin, $\epsilon_{\max}$, block start b, template $\boldsymbol{t}$, chromosome length N

`/* Subset` $\boldsymbol{\theta}[start : end]$ `to B-length subvector to update and buffers */`

$\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}[b : \min(b + B - 1, N)]$;

$\tilde{\boldsymbol{\theta}}_0 = \tilde{\boldsymbol{\theta}}$; $\underline{\boldsymbol{\theta}} = (\boldsymbol{\theta}[start : b], \boldsymbol{\theta}[\min(b + B - 1, N) : end])$;

Draw step size $\epsilon \sim \text{Unif}[\epsilon_{\min}, \epsilon_{\max}]$;

`/* Optionally estimate mass matrix from Hessian; default is identity */`

if *Estimating mass matrix*:

  Maximize log conditional posterior to obtain $\hat{\theta}$;

  $M = -\nabla_{\hat{\boldsymbol{\theta}}} \nabla_{\hat{\boldsymbol{\theta}}^\top} \log p(\hat{\boldsymbol{\theta}} | \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \underline{\boldsymbol{\theta}})$;

  if *Using diagonal mass matrix*:

    $M = \text{diag}\,(M)$;

else:

  $M = I_{end-start}$;

`/* Draw momentum */`

Draw $\boldsymbol{p} \sim N(\mathbf{0}, M)$;

$\boldsymbol{p}_0 = \boldsymbol{p}$;

`/* Run leapfrog integration */`

$\boldsymbol{p} += \epsilon \nabla_{\tilde{\boldsymbol{\theta}}} \log p(\tilde{\boldsymbol{\theta}} | \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \underline{\boldsymbol{\theta}})/2$;

for i **in** `range(`L`)`:

  $\tilde{\boldsymbol{\theta}} += \epsilon M^{-1} p$;

  if $i < L - 1$:

    $\boldsymbol{p} += \epsilon \nabla_{\tilde{\boldsymbol{\theta}}} \log p(\tilde{\boldsymbol{\theta}} | \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \underline{\boldsymbol{\theta}})$;

$\boldsymbol{p} += \epsilon \nabla_{\tilde{\boldsymbol{\theta}}} \log p(\tilde{\boldsymbol{\theta}} | \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \underline{\boldsymbol{\theta}})/2$;

`/* Metropolis-Hastings step to correct for integration errors */`

$\log r = \log p(\tilde{\boldsymbol{\theta}} | \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \underline{\boldsymbol{\theta}}) - \log p(\tilde{\boldsymbol{\theta}}_0 | \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \underline{\boldsymbol{\theta}}) - 1/2(\boldsymbol{p}^\top M^{-1} \boldsymbol{p} - \boldsymbol{p}_0^\top M^{-1} \boldsymbol{p}_0)$;

Draw u Unif$[0, 1]$;

if $u \leqslant r$:

  return $(\tilde{\boldsymbol{\theta}}, 1)$; `/* Accept update */`

else:

  return $(\tilde{\boldsymbol{\theta}}_0, 0)$; `/* Reject update */`

Algorithm 2: Worker-level HMC update

A.2.1 Choice initial estimator

We use $\hat{\theta}_k = E\left[\theta_k | \boldsymbol{y}, \hat{\mu}_{s_k}, \hat{\sigma}^2_{s_k}\right]$ as an initial point estimate of θ_k. The distributed HMC sampler presented in Section A.1 yields information on the complete marginal posterior of $\boldsymbol{\theta}$ via simulation but, given the scale of this problem, an optimization-based approach is useful as a fast initialization method. The approximate EM algorithm described in Section A.2.2 provides both approximate marginal MAP estimates of the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ and estimates of the target conditional expectations $\hat{\theta}_k$.

A.2.2 Approximate EM algorithm via Laplace approximation

We implement an approximate EM algorithm to provide initial estimates of $(\boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\sigma})$. In the E-step, we build an approximation of the conditional posterior of $\boldsymbol{\theta}$ given $(\boldsymbol{y}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}^2)$ to estimate the Q function, detailed below. The M-step updates the estimates of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ toward the marginal posterior mode of $p(\boldsymbol{\mu}, \boldsymbol{\sigma}^2 | \boldsymbol{y})$.

Approximate E-step In the E-step, the objective is to compute

$$Q_t(\boldsymbol{\mu}, \boldsymbol{\sigma}^2) = E\left[\log p(\boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2 | \boldsymbol{y}) | \boldsymbol{y}, \boldsymbol{\mu}^{(r-1)}, \boldsymbol{\sigma}^{2\,(r-1)}\right]. \tag{16}$$

The log joint posterior for $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2, \boldsymbol{\theta})$ is given by

$$\begin{aligned} \log p(\boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2 | \boldsymbol{y}, \boldsymbol{s}, \tau_0) \;=\; & -\sum_k \boldsymbol{x}_k^T \beta_k + \sum_k y_k \log\left(\boldsymbol{x}_k^T \beta_k\right) \\ & -\frac{1}{2}\sum_k \log \sigma^2_{s_k} - \frac{1}{2}\sum_k \frac{(\theta_k - \mu_{s_k})^2}{\sigma^2_{s_k}} \\ & -\frac{1}{2}\sum_s \log\left(\frac{\sigma_s^2}{n_s \tau_0}\right) - \frac{1}{2}\sum_s \frac{(\mu_s - \mu_0)^2}{\sigma_s^2 / n_s \tau_0} \\ & -\sum_s \log \sigma_s^2 + \text{const.} \end{aligned} \tag{17}$$

Thus, we can write the relevant portion of the expected log conditional posterior for $\boldsymbol{\theta}$ given $\{\mu_{s_k}, \sigma^2_{s_k}\}$ as

$$\begin{aligned} Q_t(\boldsymbol{\mu}, \boldsymbol{\sigma}^2) \;=\; & -\frac{1}{2}\sum_k \log \sigma^2_{s_k} - \frac{1}{2}\sum_k \frac{(\hat{\theta}_k - \mu_{s_k})^2}{\sigma^2_{s_k}} - \frac{1}{2}\sum_k \frac{\hat{V}_k}{\sigma^2_{s_k}} \\ & -\frac{1}{2}\sum_s \log\left(\frac{\sigma_s^2}{n_s \tau_0}\right) - \frac{1}{2}\sum_s \frac{(\mu_s - \mu_0)^2}{\sigma_s^2 / n_s \tau_0} - \sum_s \log \sigma_s^2. \end{aligned} \tag{18}$$

where $\hat{\theta}_k = E[\theta_k|\boldsymbol{\mu}_{(t-1)}, \boldsymbol{\sigma}^2_{(t-1)}]$ and $\hat{V}_k = \text{Var}[\theta_k|\boldsymbol{\mu}_{(t-1)}, \boldsymbol{\sigma}^2_{(t-1)}]$.

While the conditional posterior $p(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ is available in close form, the necessary expectations $\hat{\theta}_k$ and variances $\hat{V}_k$ are not. However, under the proposed log-Normal/Poisson model structure, the univariate conditional posteriors of θ_k given $\{\mu_{s_k}, \sigma^2_{s_k}\}$ are unimodal, log-concave, nearly symmetric, and have tails that go to zero as $\exp(-c\theta_k^2)$. Thus, these conditional posteriors are nearly Gaussian and a Laplace approximation is appropriate.

To compute the Laplace approximation, we first find the posterior mode of θ_k given $(\boldsymbol{\mu}^{(r-1)}, \boldsymbol{\sigma}^{2\,(r-1)})$. This amounts to maximizing

$$g(\boldsymbol{\theta}) = -\sum_k \boldsymbol{x}_k^T \beta_k + \sum_k y_k \log\left(\boldsymbol{x}_k^T \beta_k\right) - \frac{1}{2}\sum_k \frac{(\theta_k - \mu_{s_k})^2}{\sigma^2_{s_k}} \tag{19}$$

with respect to $\boldsymbol{\theta}$. This mode is not available in closed form, but the given objective function is concave and has a continuous gradient, so numerical optimization is feasible.

The Laplace approximation then consists of substituting a Gaussian distribution with mean $\hat{\theta}_t$ equal to the mode of g, and variance $\hat{V}_k = -\text{diag}\,(H^{-1})_k$ for the conditional posterior $p(\theta_k|\boldsymbol{y}, \mu_{s_k}, \sigma^2_{s_k})$.

M-step The M-step consists of maximizing $Q_t(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ with respect to μ_{s_k} and $\sigma^2_{s_k}$. We obtain two simple closed-form solutions, summarized in Equations 20 and 21:

$$\hat{\mu}_s = \frac{1}{1+\tau_0}\left(\frac{1}{n_s}\sum_{k:s_k=s}\hat{\theta}_k\right) + \frac{\tau_0}{1+\tau_0}\mu_0 \tag{20}$$

$$\hat{\sigma}^2_s = \frac{\frac{1}{n_s}\sum_{k:s_k=s}(\hat{\theta}_k - \hat{\mu}_s)^2 + \frac{1}{n_s}\sum_{k:s_k=s}\hat{V}_k + \tau_0(\hat{\mu}_s - \mu_0)^2}{1+\tau_0+2/n_s} \tag{21}$$

The term $\hat{V}_k$ differentiates the M-step update of σ_s from the update obtained from joint maximization of the log-posterior. The joint mode of this log-posterior is reached at $\boldsymbol{\sigma}^2 = \mathbf{0}$ and $\theta_k = \mu_{s_k}\,\forall k$, as these values would allow the joint log-posterior density to increase without bound. Algorithmically, the $\hat{V}_k$ term introduced by the EM algorithm prevents $\hat{\sigma}^2$ from collapsing to 0, providing non-degenerate inferences.

A.2.3 Distributed approximate E-step

We use the conditional independence structure of $\boldsymbol{\beta}$ given $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ and partitions discussed in Section A.1.2 to distribute our approximate E-step across multiple processors. Given each partition of $\boldsymbol{\theta}$, we update the Laplace approximations in parallel, by finding the mode of the conditional posterior subvector-by-subvector. The overall algorithm is blockwise coordinate

ascent within each approximate E-step, with each E-step consisting of iterative maximization of $\log p(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ using different subsets of $\boldsymbol{\theta}$ (corresponding to different blocks) in each iteration. This converges to the maximum of $g(\boldsymbol{\theta})$ given $\boldsymbol{\mu}^{(r-1)}$ and $\boldsymbol{\sigma}^{2\,(r-1)}$.

The approximate E-step considers each block in each of the four configurations, during one iteration. More details are give in Section A.2.4. Within each block $m_1 : m_2$, we maximize $\log p(\boldsymbol{\theta}_{m_1:m_2}|\boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \boldsymbol{\theta}_{-(m_1:m_2)})$ numerically via L-BFGS-B or a truncated Newton algorithm (Zhu et al., 1997; Nash, 2000); the latter is typically more efficient in this application. We carry out this maximization directly, avoiding the data augmentation typically used in additive Poisson models of this type (e.g., van Dyk et al., 2006). Such data augmentation would require storing and computing at least $(2w+1)N$ additional variables, provide slower convergence, and slow the overall computation substantially. By controlling the size of the blocks, we can keep the scale of each optimization problem small enough that direct numerical maximization of these conditional posteriors is not a limiting factor for the algorithm (less than 100ms, typically).

We compute the Hessian of the conditional log-posterior $\log p(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ after each complete scan through the partitions, completing the approximate E-step and providing the information necessary for our M-step. The Hessian is sparse, but its inversion is computationally-intensive even with modern sparse-matrix solvers. Thus, we typically use a diagonal approximation to the Hessian. The diagonal approximation works well in our setting, even though one would expect the strong local dependence generated by the digestion matrix to produce a Hessian with large off-diagonal elements. However, due to the use of exchangeable local regularization, the Hessian is typically diagonally-dominant. The diagonal approximation is quite accurate; we observed few differences to 2-3 significant digits in comparisons of the estimated Hessian and its inverse on small portions of the genome (single ORFs with promoters, approximately 1,000 to 10,000bp in length). We lay out the overall structure of this approximate EM algorithm in Algorithm 3.

Outline of Approximate EM Algorithm
while *not converged*:
`/* E-step */`
for *Partition* **in** (D_1, D_2):
Update $\hat{\boldsymbol{\theta}}$ via numerical maximization of $g(\boldsymbol{\theta})$ within each subvector
Approximate Var $\left[\theta_k|\boldsymbol{y}, \mu_{s_k}, \sigma^2_{s_k}\right]$ by inverting Hessian of $p(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$
`/* M-step */`
Update $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ as maximizers of $E\left[p(\boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2|\boldsymbol{y}, \boldsymbol{s}, \tau_0)|\boldsymbol{\mu}_{(t-1)}, \boldsymbol{\sigma}^2_{(t-1)}, \boldsymbol{s}, \boldsymbol{\tau}_0\right]$

Algorithm 3: Approximate EM Algorithm

A.2.4 Algorithmic details

The algorithm outlined in Section A.2.2 can be implemented on distributed systems with MPI, using the same techniques as the MCMC algorithm presented in Section A.1. Due to the use of a quasi-Newton optimization algorithm within each worker's approximate E-step, its computational complexity scales $O(B^2)$ for each such update. However, it scales $O(N)$ in the length of genome given a fixed block size B. Memory requirements are $O(N)$ for the master process running the M-step and coordinating the approximate E-step and $O(B^2)$ (independent of N) for the worker processes running the approximate E-step updates. We lay out the details of this parallel approximate E-step in Algorithm 4.

Parallel Implementation of Approximate E-Step

`/* Send conditioning information */`
Broadcast $\boldsymbol{\mu}^{(t-1)}$, and $\boldsymbol{\sigma}^{2\,(t-1)}$ to all workers ;
for *offset* **in** *(0, B/2)*:
  `/* Send first round of jobs to workers */`
  for w **in** `range(`*min(nWorkers, nBlocks)*`)`:
    $start = \max(0, (w-1)(B+2w) - 2w + \text{offset}) + 1$;
    $end = \min(N, w(B+2w) + 2w + \text{offset})$;
    Send $\boldsymbol{\theta}[start : end]$ to worker process w with work tag attached ;
  if *len(startVec)* $<$ *nWorkers*:
    Pause remaining workers ;
  `/* Collect results */`
  Set *nComplete* $= 0$, *nStarted* $= \min(\text{nWorkers}, \text{nBlocks})$
  while *nComplete* $<$ *nBlocks*:
    Receive result $\boldsymbol{\theta}[start : end]$ from arbitrary worker with tag b_1 ;
    Incorporate result into working copy of $\boldsymbol{\theta}^{(t)}$;
    nComplete++ ;
    if *nStarted* $<$ *nBlocks*:
      `/* Send additional jobs as needed */`
      $w = nStarted + 1$;
      $start = \max(0, (w-1)(B+2w) - 2w + \text{offset}) + 1$;
      $end = \min(N, w(B+2w) + 2w + \text{offset})$;
      Send $\boldsymbol{\theta}[start : end]$ to last completed worker process with work tag attached;
      nStarted++;
`/* Compute approximate variance, if needed */`
Compute approximate $\text{Var}\left[\theta_k | \boldsymbol{y}, \mu_{s_k}, \sigma^2_{s_k}\right]$ using sparse Cholesky decomposition or diagonal approximation to Hessian;

Algorithm 4: Approximate E-Step

We this algorithm process a chromosome with $1.5e6$ base pairs in only 11 minutes using 256 threads on Amazon EC2; a chromosome with $2.4e5$ base pairs requires only 1 minute.

This method will easily scale to genomes of far greater size (e.g. mice) with this distributed structure, especially using resources such as EC2. The one-to-one substitution of time and processors possible on the cloud makes it an ideal infrastructure for running this type of method.

B Additional figures

B.1 Reproducibility analysis—comparability of cluster-level estimators

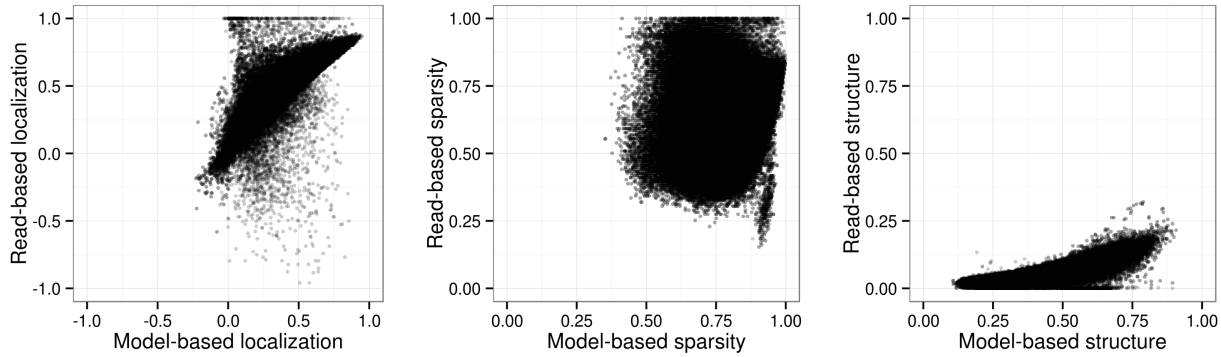

Figure 1: Joint distributions of of model-based and read-based estimates of cluster-level properties.

B.2 Power analysis—cluster locations

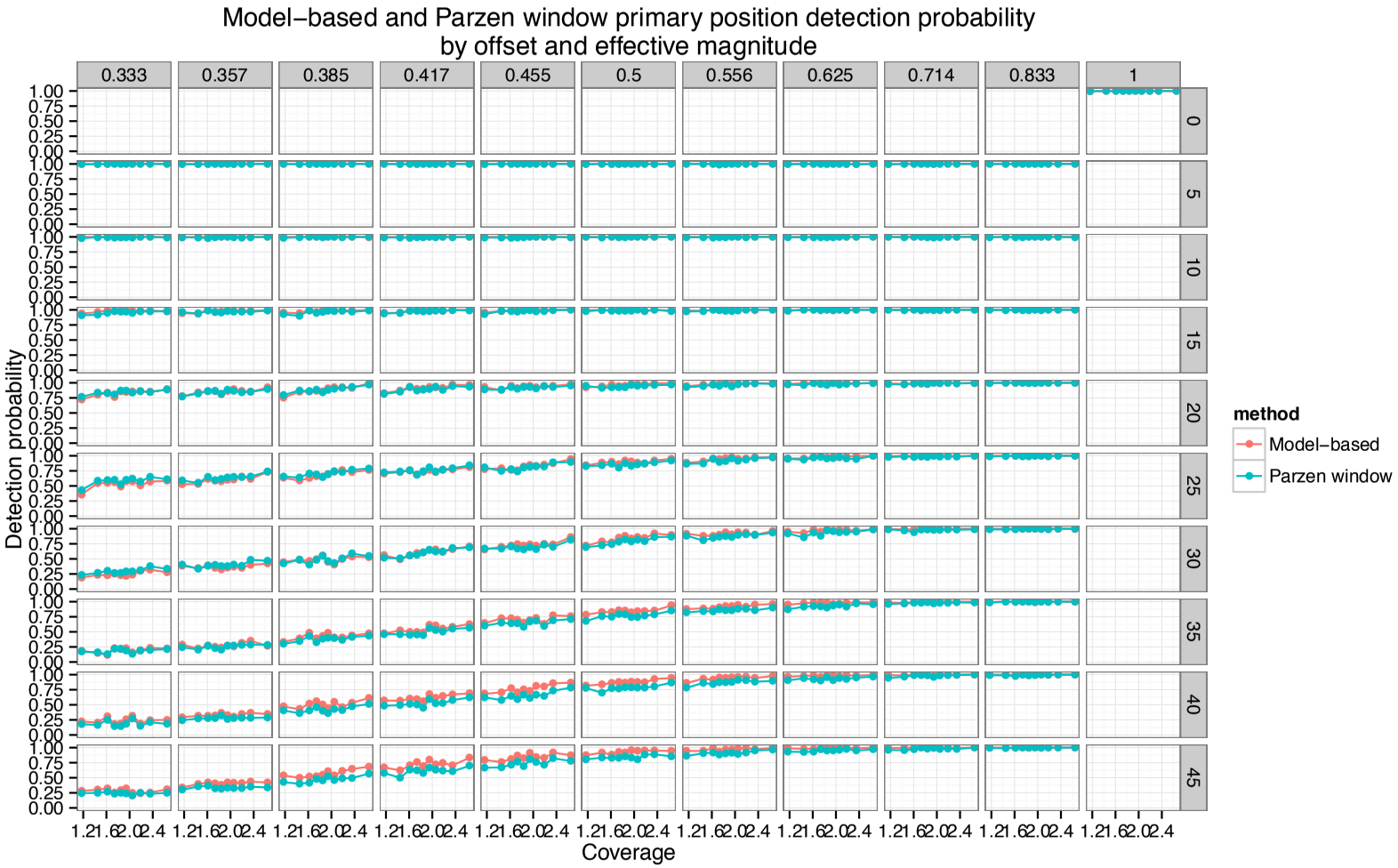

Figure 2: Power of model-based and Parzen window methods to detect cluster centers ±5bp vs. coverage by alternative position offset (rows) and effective magnitude of primary position (columns)

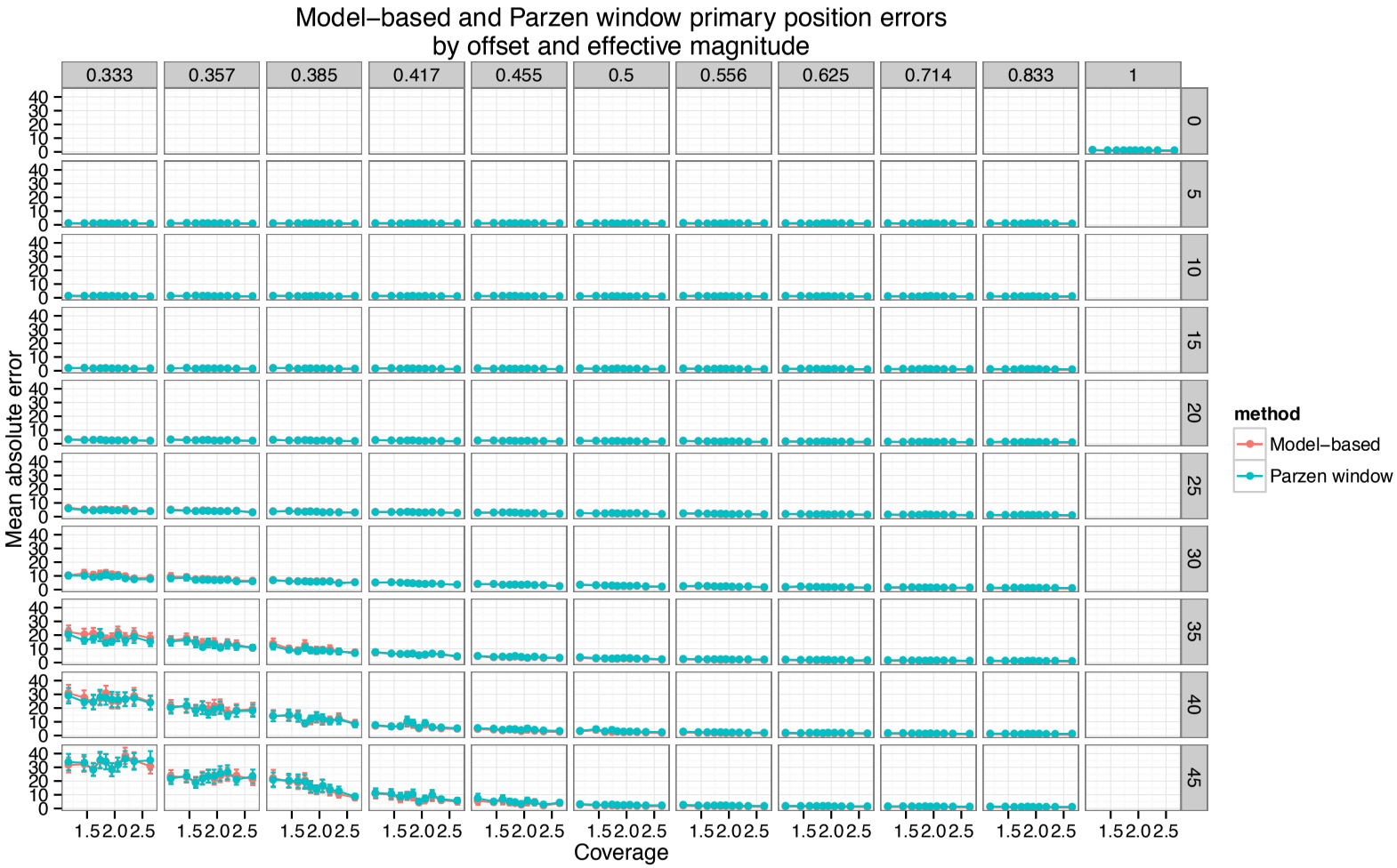

Figure 3: Mean absolute position errors for model-based and Parzen window methods vs. coverage by alternative position offset (rows) and effective magnitude of primary position (columns)

B.3 Power analysis—local concentrations

B.3.1 Primary positions

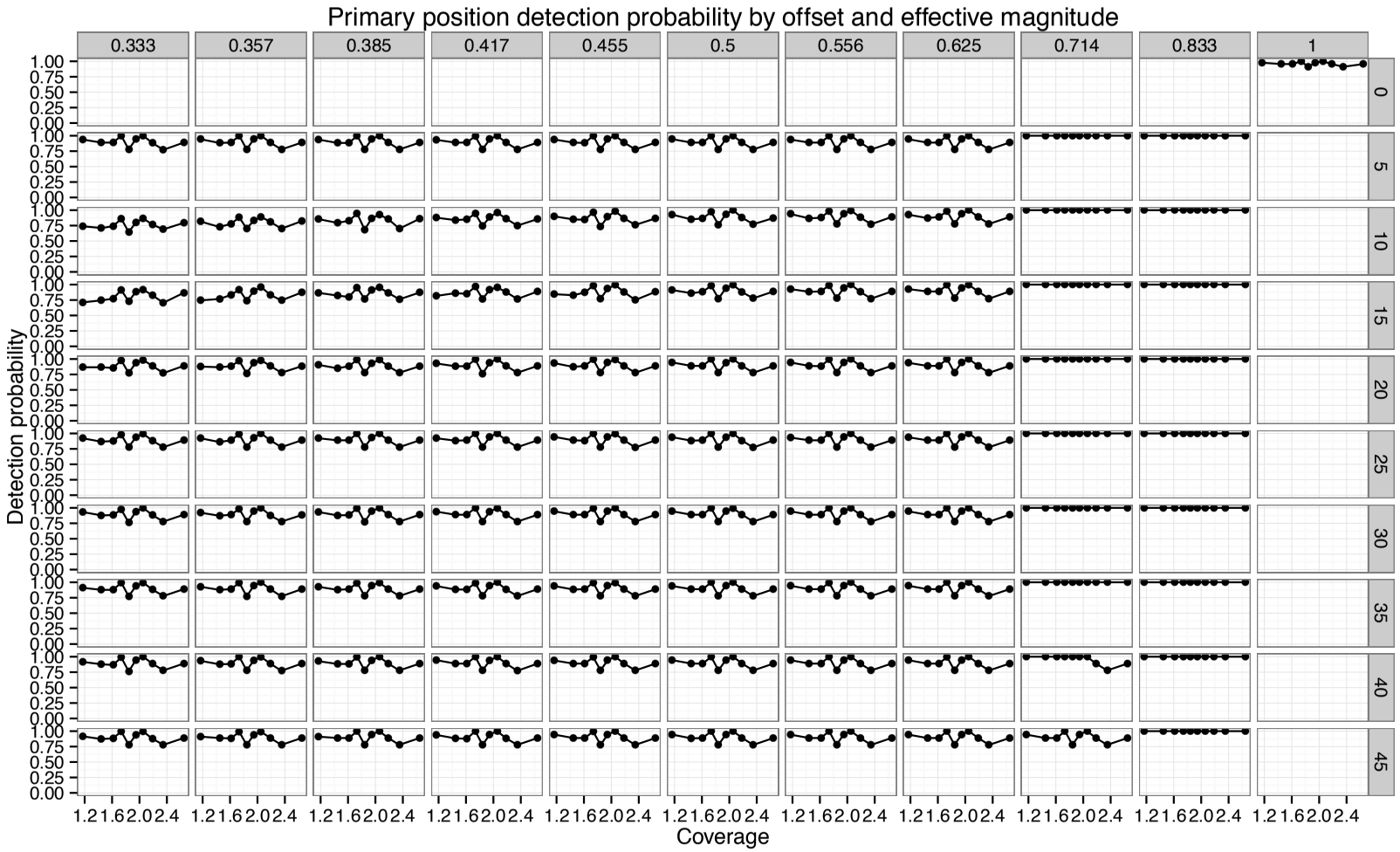

Figure 4: Power of model-based method to detect individual primary positions ±5bp vs. coverage by alternative position offset (rows) and effective magnitude of primary position (columns)

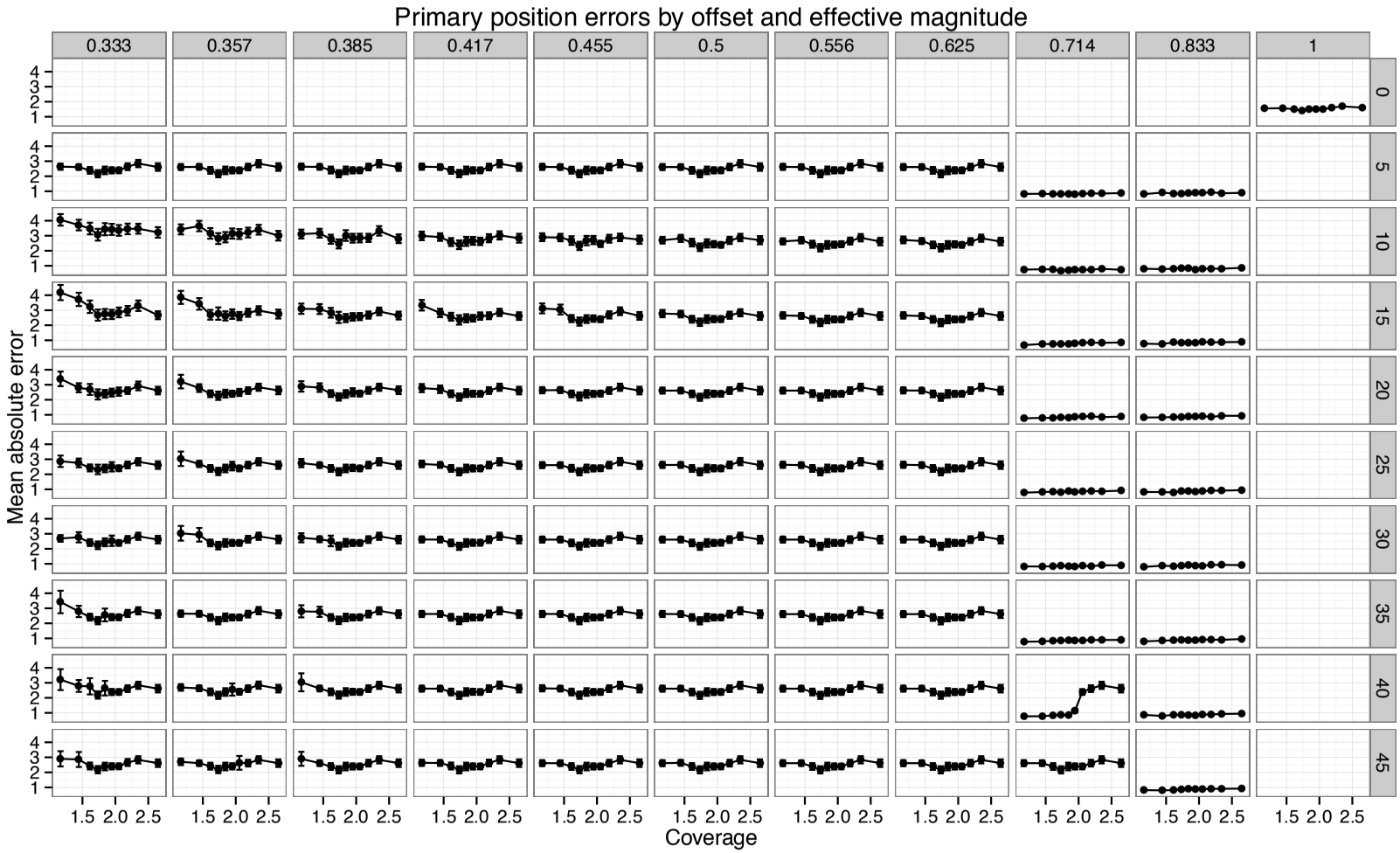

Figure 5: Mean absolute position errors of model-based method for individual primary positions vs. coverage by alternative position offset (rows) and effective magnitude of primary position (columns)

B.3.2 Alternative positions

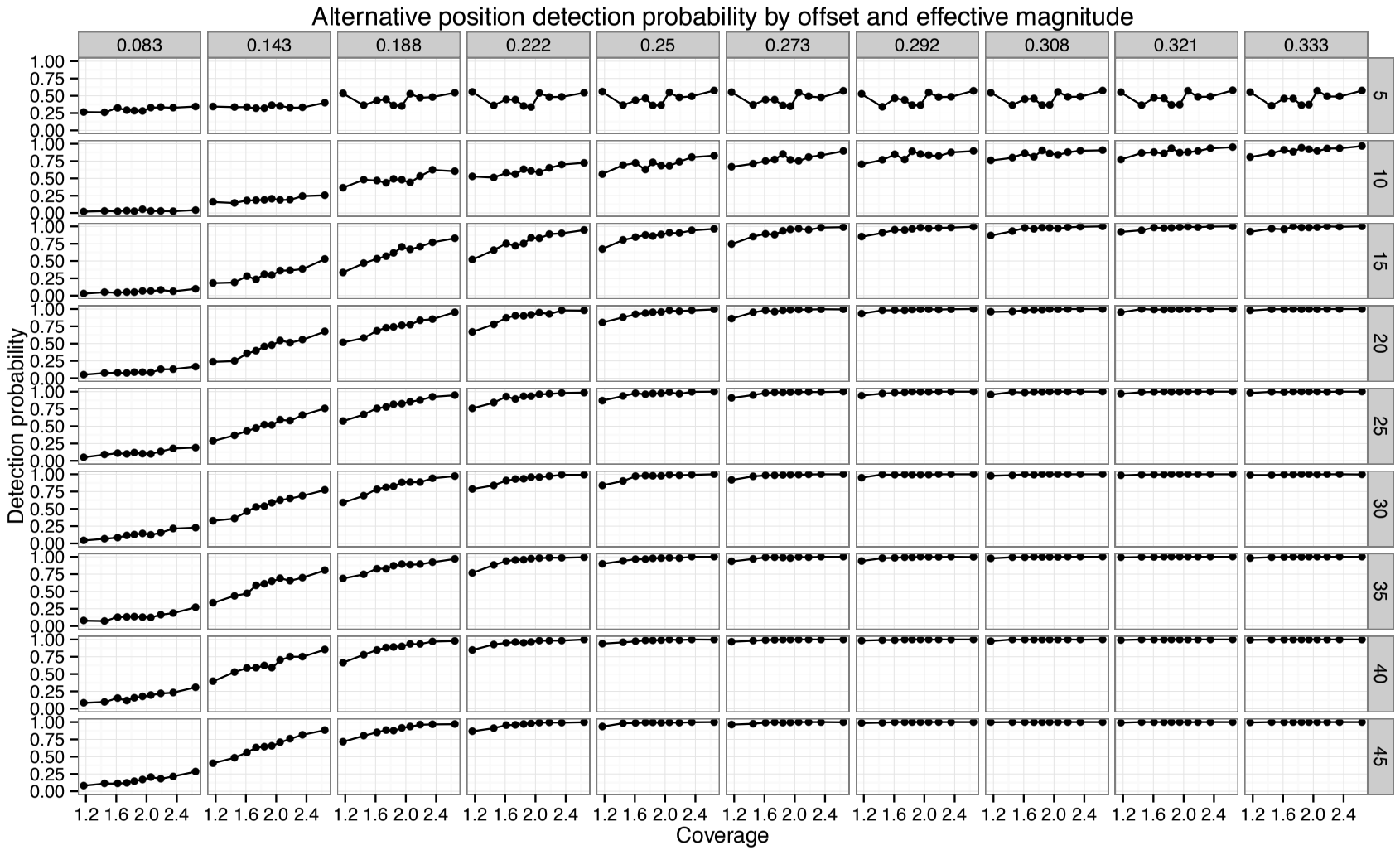

Figure 6: Power of model-based method to detect individual alternative positions ±5bp vs. coverage by alternative position offset (rows) and effective magnitude of alternative position (columns)

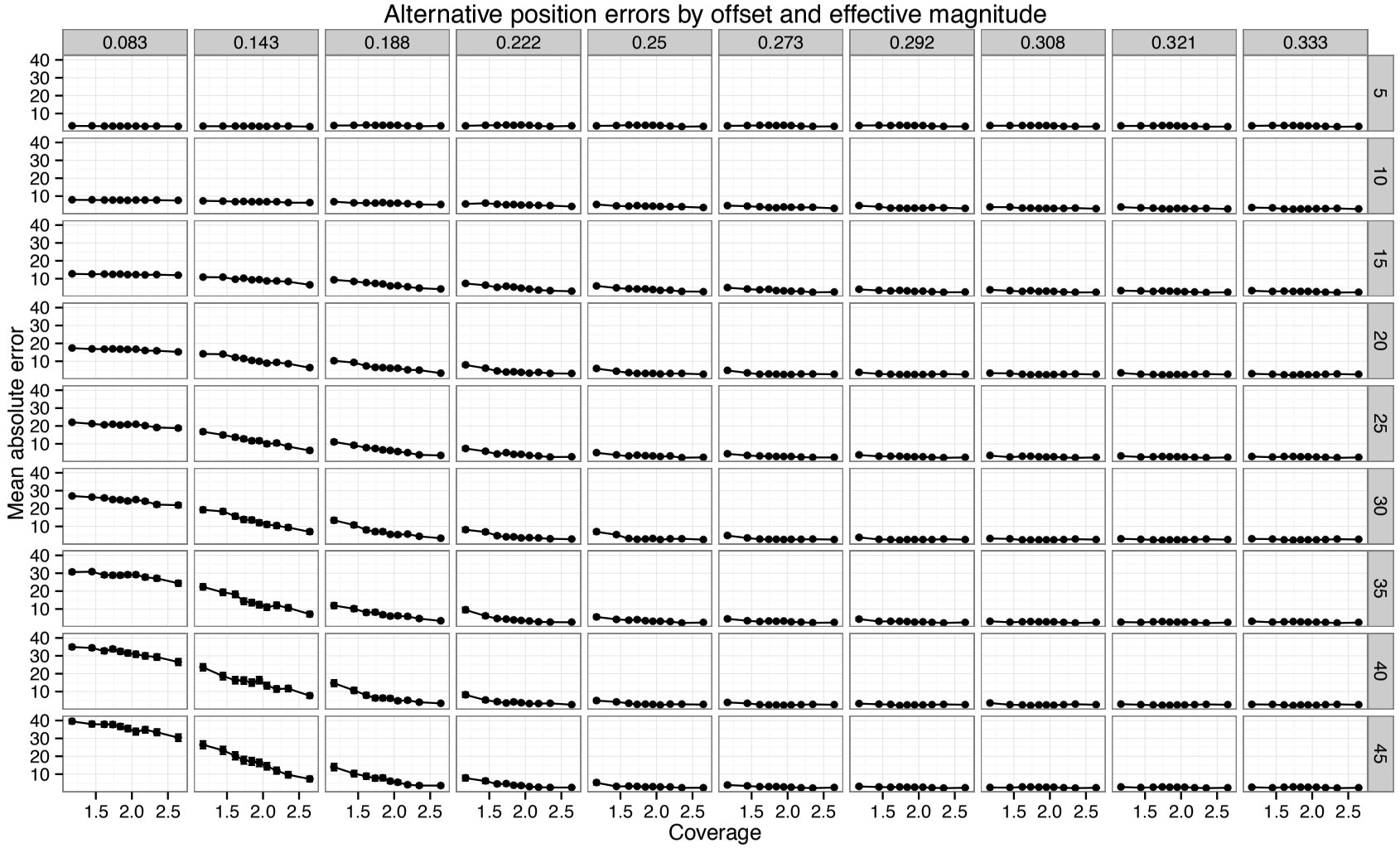

Figure 7: Mean absolute position errors of model-based method for individual alternative positions vs. coverage by alternative position offset (rows) and effective magnitude of alternative position (columns)

References

S.G. Nash. A survey of truncated-Newton methods. *Journal of Computational and Applied Mathematics*, 124(1):45–59, 2000.

R.M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 54:113–162, 2010.

D.A. van Dyk, A. Connors, D.N. Esch, P. Freeman, H. Kang, M. Karovska, V. Kashyap, A. Siemiginowska, and A. Zezas. Deconvolution in high-energy astrophysics: Science, instrumentation, and methods. *Bayesian Analysis*, 1(2):189–236, 2006.

Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw.*, 23:550–560, December 1997. ISSN 0098-3500. doi: http://doi.acm.org/10.1145/279232.279236. URL http://doi.acm.org/10.1145/279232.279236.