This is the title of the thesis

A dissertation presented
by
Alexander Weaver Blocker
to
The Department of Statistics
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of
Statistics

Harvard University
Cambridge, Massachusetts
May 2013

© *2013 -Alexander Weaver Blocker*
All rights reserved.

Professor Xiao-Li Meng Alexander Weaver Blocker

This is the title of the thesis

ABSTRACT

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.

Quisque facilisis erat a dui. Nam malesuada ornare dolor. Cras gravida, diam sit amet rhoncus ornare, erat elit consectetuer erat, id egestas pede nibh eget odio. Proin tincidunt, velit vel porta elementum, magna diam molestie sapien, non aliquet massa pede eu diam. Aliquam iaculis. Fusce et ipsum et nulla tristique facilisis. Donec eget sem sit amet ligula viverra gravida. Etiam vehicula urna vel turpis. Suspendisse sagittis ante a urna. Morbi a est quis orci consequat rutrum. Nullam egestas feugiat felis. Integer adipiscing semper ligula. Nunc molestie, nisl sit amet cursus convallis, sapien lectus pretium metus, vitae pretium enim wisi id lectus. Donec vestibulum. Etiam vel nibh. Nulla facilisi. Mauris pharetra. Donec augue. Fusce ultrices, neque id dignissim ultrices, tellus mauris dictum elit, vel lacinia enim metus eu nunc.

Proin at eros non eros adipiscing mollis. Donec semper turpis sed diam. Sed consequat ligula nec tortor. Integer eget sem. Ut vitae enim eu est vehicula gravida. Morbi ipsum ipsum, porta nec, tempor id, auctor vitae, purus. Pellen-

tesque neque. Nulla luctus erat vitae libero. Integer nec enim. Phasellus aliquam enim et tortor. Quisque aliquet, quam elementum condimentum feugiat, tellus odio consectetuer wisi, vel nonummy sem neque in elit. Curabitur eleifend wisi iaculis ipsum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. In non velit non ligula laoreet ultrices. Praesent ultricies facilisis nisl. Vivamus luctus elit sit amet mi.

Contents

1 Chapter 1 title 1

2 Template-based estimation of genome-wide nucleosome positioning via distributed HMC 4

- 2.1 Introduction 4
  - 2.1.1 Related work 5
  - 2.1.2 Contributions of this article 6
- 2.2 Model 7
  - 2.2.1 Digestion variability template 9
  - 2.2.2 Segmentation 11
  - 2.2.3 Estimands 13
- 2.3 Inference, estimation, and computation 14
  - 2.3.1 Template estimation 15
  - 2.3.2 Segmentation algorithm 17
  - 2.3.3 Posterior sampler 18
  - 2.3.4 Detection and calibration 20
- 2.4 Results 21
  - 2.4.1 Experimental design 22
  - 2.4.2 Parallel HMC performance 24
  - 2.4.3 Power analysis 25
  - 2.4.4 Reproducibility analysis 31
- 2.5 Concluding Remarks 35
  - 2.5.1 Modeling 35
  - 2.5.2 Estimands 36
  - 2.5.3 Inference 37

A Online supplement for "Template-based estimation of genome-wide nucleosome positioning via distributed HMC" 46
A.1 Algorithmic details of inference 46
A.1.1 Distributed HMC sampler 46
A.1.2 Approximate EM algorithm 51
A.2 Additional figures 57
A.2.1 Reproducibility analysis—comparability of cluster-level estimators 57
A.2.2 Power analysis—cluster locations 57
A.2.3 Power analysis—local concentrations 64

THIS IS THE DEDICATION.

AUTHOR LIST

The following authors contributed to Chapter 1: Alexander W Blocker.

The following authors contributed to Chapter 2: Alexander W Blocker, Edoardo M Airoldi.

The following authors contributed to Chapter 3: Alexander W Blocker.

ACKNOWLEDGMENTS

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.

Nulla facilisi. In vel sem. Morbi id urna in diam dignissim feugiat. Proin molestie tortor eu velit. Aliquam erat volutpat. Nullam ultrices, diam tempus vulputate egestas, eros pede varius leo.

Quoteauthor Lastname

1

Chapter 1 title

THERE'S SOMETHING TO BE SAID for having a good opening line. Morbi commodo, ipsum sed pharetra gravida, orci $x = 1/\alpha$ magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.

$$\zeta = \frac{1039}{\pi}$$

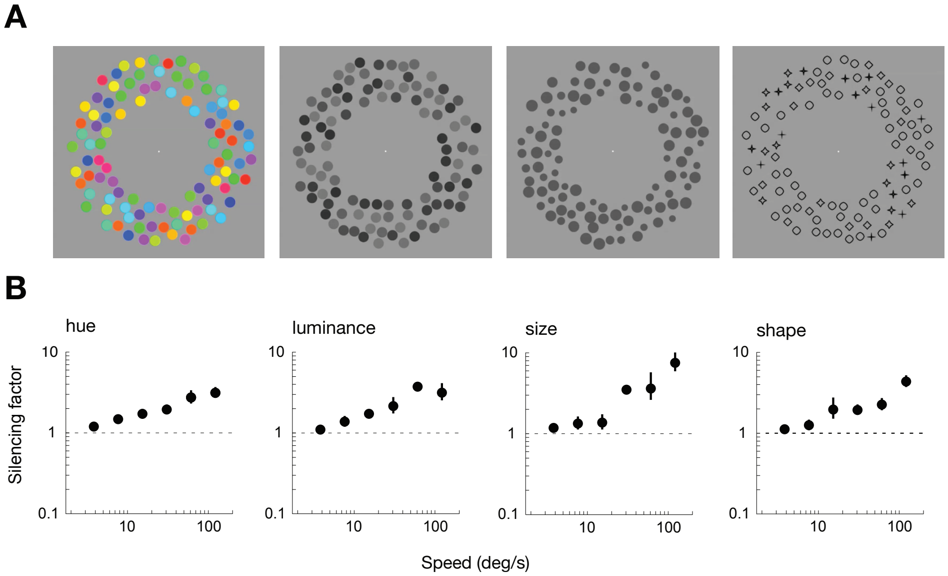
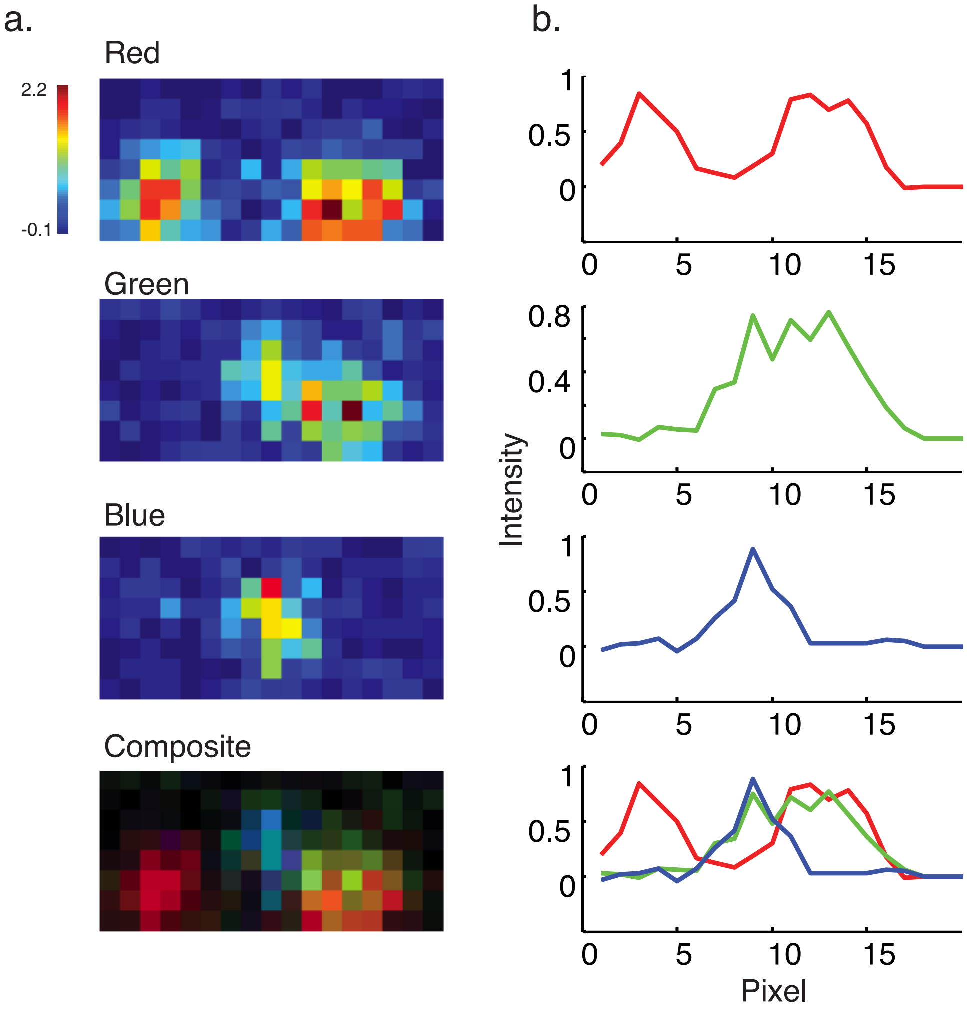


Figure 1.1: This is a figure that floats inline and here is its caption.

Figure 1.2 ***(following page)*****:** This is a full page figure using the FPfigure command. It takes up the whole page and the caption appears on the preceding page. Its useful for large figures. Harvard's rules about full page figures are tricky, but you don't have to worry about it because we took care of it for you. For example, the full figure is supposed to have a title in the same style as the caption but without the actual caption. The caption is supposed to appear alone on the preceding page with no other text. You do't have to worry about any of that. We have modified the fltpage package to make it work. This is a lengthy caption and it clearly would not fit on the same page as the figure. Note that you should only use the FPfigure command in instances where the figure really is too large. If the figure is small enough to fit by the caption than it does not produce the desired effect. Good luck with your thesis. I have to keep writing this to make the caption really long. LaTex is a lot of fun. You will enjoy working with it. Good luck on your post doctoral life! I am looking forward to mine.

Figure 1.2: (continued)



This is some random quote to start off the chapter.

Firstname lastname

2

Template-based estimation of genome-wide nucleosome positioning via distributed HMC

2.1 INTRODUCTION

The organization of genetic material within cells plays a major role in the regulation of biological activities. In the cell, DNA is wrapped around histone proteins to form nucleosomes, which constitute the smallest units of such 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 paper, 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., 2007a). 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 consists of the number of read centers that align to each base pair along the genome. 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. Figure 2.1 illustrates some example data.

2.1.1 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 for the analysis 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., 2007b; Shivaswamy et al., 2008; Tsankov et al., 2010). More recent studies have used paired-end sequencing technology (Gkikopoulos et al., 2011).

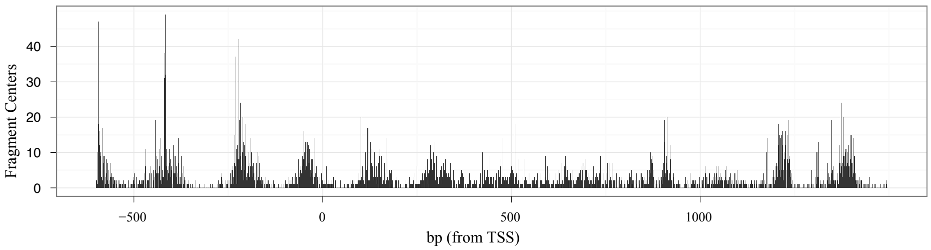


Figure 2.1: Example data for yeast gene PHO5.

The statistical approach to identifying nucleosome positions form tiling microarray data consisted largely of hidden Markov models and their variants (Gupta, 2007; Yuan and Liu, 2008; Yassour et al., 2008; Sun et al., 2009b; Mitra and Gupta, 2011), with some mixture model approaches also in use (Sun et al., 2009a). Most analyses of sequencing data have adopted Parzen-window based estimators, which convolve the observed read counts within a window, extract local maxima, and perform subsequent computations based on 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). Others have adapted HMMs to this class of data (Cairns et al., 2011). Model-based analyses of sequencing data have focused on mixture models (Polishko et al., 2012; Rashid et al., 2011; Zhang et al., 2012). Recent work combines a new biochemical protocol with a Bayesian deconvolution method (Brogaard et al., 2012); however, their inference procedure targets different estimands.

Methodology relevant to the problem we consider has been developed to infer transcription-factor (TF) 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., 2011), a Poisson model for sequence counts (Zhang et al., 2008), and detection methods for peak finding (Pepke et al., 2009). From a statistical perspective, a key feature of ChIP-seq data is that the TF binding sites are non-overlapping. This allows for independence assumptions in models for ChIP-seq data (Barski and Zhao, 2009) that cannot be defended in models of nucleosomes, which are likely to overlap when cell populations are sequenced.

2.1.2 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 2.2. 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 sampler to draw from the posterior distribution of the quantities of interest under our model, in Section 2.3. This sampler is highly amenable to distributed computation and scales linearly with the length of the genome being analyzed. We provide a non-parametric 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 Section 2.4, 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.

2.2 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 $p(\vec{y}|\vec{\beta})$, which provides the distribution of the observed read counts given the underlying distribution of nucleosome positions $\vec{\beta}$, and a positioning model $p(\vec{\beta}|\vec{\mu}, \vec{\sigma}^2)$, which describes the structure of the nucleosome position distribution. Given a segmentation function, $s : \{1 \ldots N\} \rightarrow \{1 \ldots S\}$, which maps the N base pair locations to S regions in which coefficients β_k can be assumed to be identically distributed, we posit

$$
\begin{aligned}
y_k|\lambda_k &\sim \text{Poisson}\left(\lambda_k\right) && (2.1)\\
\vec{\lambda}_{(N\times 1)} &\equiv X_{(N\times(N-2\lfloor \ell_0/2\rfloor))}\ \vec{\beta}_{((N-2\lfloor \ell_0/2\rfloor)\times 1)}, && (2.2)\\
&\qquad \beta_k > 0 \text{ for } k = \lfloor \ell_0/2\rfloor + 1\ldots N - \lfloor \ell_0/2\rfloor\\
\log\beta_k &\sim \text{Normal}\left(\mu_{s_k}, \sigma^2_{s_k}\right) && (2.3)
\end{aligned}
$$

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, and $s(k)$ is denoted as s_k for compactness. (The construction of the matrix X is detailed in Section 2.2.1.) The log-likelihood for the proposed model is as follows, subject to the positivity constraint on $\vec{\beta}$,

$$
\begin{aligned}
\log p(\vec{y}|\vec{\theta},\vec{\mu},\vec{\sigma}^2) \quad = \quad & -\sum_k \vec{x}_k^T\vec{\beta} + \sum_k y_k \log\left(\vec{x}_k^T\vec{\beta}\right) && (2.4)\\
& -\frac{1}{2}\sum_k \log\sigma^2_{s_k} - \frac{1}{2}\sum_k \frac{(\theta_k - \mu_{s_k})^2}{\sigma^2_{s_k}} + \text{const.}
\end{aligned}
$$

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. These stabilize our inferences and reflect vague prior information on the distribution of $\vec{\beta}$. 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 2.3. We analyze the sensitivity of the inferences to the choice of (μ_0, τ_0, α_0, γ_0) in Section 2.4.

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 subsections, before introducing the estimands of interest in Section 2.2.3.

2.2.1 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 A.2. We cover the paired-end case here and discuss extensions to single-end sequencing in Section 2.5. Consider a simple model for the variability of enzymatic digestion. We denote the length of each fragment j as ℓ_j and assume

$$\ell_j = \ell_0 + e_{1j} + e_{2j}, \quad e_{1j}, e_{2j} \sim IID\ . \tag{2.5}$$

ℓ_0 is the base length of each fragment and the $e_{\cdot j}$ terms are the digestion errors at each end of the fragment. We assume these errors are bounded and symmetric between the ends of each fragment; physically, this means that the enzyme has the same propensity towards 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,

$$t_k = P(d_j = k) + \frac{1}{2}\left(P(d_j = k - \frac{1}{2}) + P(d_j = k + \frac{1}{2})\right) \tag{2.6}$$

for $k = -w, \ldots, w$, yielding a vector $\vec{t}$ of length $2w + 1$. We estimate the template from the empirical fragment length distribution corresponding to a lane of paired-end sequencing data, as detailed in Section 2.3.1. Example exact and approximate templates for the same data are shown in Figure 2.2.

The matrix X in Equation A.2 is constructed using a template $\vec{t}$ by leveraging an equivalence between a realistic data-generating process and the marginal specification given in Equations A.1–A.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 Equa-

Figure 2.2: Example templates for yeast growing in high-phosphate.

tions A.1–A.2.

In detail, 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 an $(N \times N - \ell_0)$ matrix where each row corresponds to a shifted version of the template. The matrix X is fully specified as follows,

$$X = \begin{pmatrix} \vec{t} & & & & \\ & \vec{t} & & & \\ & & \ddots & & \\ & & & \vec{t} & \\ & & & & \vec{t} \end{pmatrix} \tag{2.7}$$

Using the $(N - \ell_0)$-dimensional constrained vector of coefficients $\vec{\beta} \geq 0$, we obtain an N-dimensional vector of expected counts $\vec{\lambda}$ using Equation A.2. Each coefficient β_k provides the number of fragments we expect to sequence from nucleosomes centered at position k. Analogously, $\vec{\lambda}$ provides the number of fragment centers we expect to observe at each position. Formally, $\vec{\lambda}$ is a convolution of $\vec{\beta}$ with $\vec{t}$. 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-lognormal variation) and "horizontal" convolution across the sequence (from digestion variability) creates a challenging deconvolution problem.

2.2.2 SEGMENTATION

Our segmentation of the DNA sequence accounts 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 appear sensible. 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 2.3.

Statistically, the segmentation enables local regularization in the estimation of $\vec{\beta}$. These coefficients are weakly identified in a model specified by Equations A.1–A.2 alone. Such a model would involve $N - \ell_0$ parameters and N observations, and the Hessian matrix for the implied log-likelihood of $\vec{\beta}$ would be $H = -X^T W X$, where $W = \operatorname{diag}\left(y_1/\vec{\lambda}_1^2, \ldots, y_n/\vec{\lambda}_n^2\right)$. This is negative-definite if $\vec{y}$ contains no all-zero subvector of length $2w+1$ or more; otherwise, it is only negative semi-definite. Furthermore, H is typically very 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 A.3. In this complete model, we pool information locally within each 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

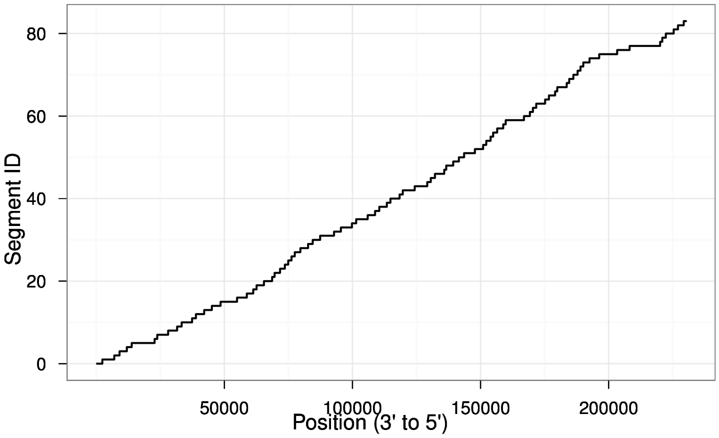


Figure 2.3: Example segmentation of yeast chromosome I. See Section 2.3 for estimation details.

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 have a very low concentration of nucleosome positions. Such locations correspond to small, non-zero 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 such behavior: it allows the majority of values in $\vec{\beta}$ 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 $\vec{\beta}$, 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; however, 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 a strategy to managing this trade-off in Section 2.3.2.

2.2.3 ESTIMANDS

We can express the scientific estimands of interest as functions of $\vec{\beta}$. The parameter $\vec{\beta}$ itself is of interest, as it captures the pattern of nucleosome positioning across each chromosome. However, $\vec{\beta}$ is high-dimensional and unsuitable for human interpretation. The posterior expectation, standard deviation, and quantiles of $\vec{\beta}$ are useful for visualization and exploratory analysis. Below, we develop several more refined estimands to quantify the structure of the nucleosome position distribution. These new 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 $\vec{\beta}$, we consider the ratio

$$C_{p,l}(k) = \frac{\sum_{i=-p}^{p} \beta_{k+i}}{\sum_{i=-l}^{l} \beta_{l+i}} \,, \tag{2.8}$$

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 $\vec{\beta}$ 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 147bp of DNA wrapped around histone proteins, so, within a single cell, nucleosomes must be spaced by at least 147bp. As a result, each cell can contribute at most one nucleosome center within a window of width 147bp or less. Thus, the relative magnitudes of the entries of $\vec{\beta}$ 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. However, choosing $p > 0$ is generally preferred to account for 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 $C_{p,l}(k) = \frac{p}{l}$. 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 A.2.3.

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, such as Parzen window filtering, applied to the estimated vector of $\vec{\beta}$ coefficients. Because of this, these estimands may inherit issues from this clustering procedures; however, they are useful for comparative analysis and can capture interesting patterns. We define the estimand $\vec{\kappa}$ to be the cluster centers obtained by running the selected clustering method on $\vec{\beta}$. $\vec{\kappa}$ is a cluster-level estimand itself, but it is primarily of interest as a means to obtain summaries of $\vec{\beta}$ 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 $\vec{\beta}$, taken as an unnormalized discrete distribution over the base pairs in the cluster. Formally, considering cluster $\vec{\beta}_{[i:j]}$ and defining $p_{[i,j]}(k) = \beta_k / \sum_{m=i}^{j} \beta_m$, our localization, structure, and sparsity measures are defined as

$$\begin{aligned}
L_{i,j} &= 1 - \frac{4}{j-i+1} \sum_{k=i}^{j} p_{[i,j]}(k) |k - m_{i,j}|, \; m_{i,j} = \sum_{k=i}^{j} k p_{[i,j]}(k) && (2.9) \\
S_{i,j} &= 1 + \frac{1}{\log(j-i+1)} \sum_{k=i}^{j} \log p_{[i,j]}(k) && (2.10) \\
R_{i,j,q} &= 1 - \frac{n_{i,j,q} - 1}{q(j-i+1)}, \; n_{i,j,q} = \min \left(n : \sum_{k=i}^{i+n} \tilde{p}_{[i,j]}(k) \right), && (2.11)
\end{aligned}$$

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 $\vec{\beta}_{[i:j]}$ is constant and $L_{i,j} = S_{i,j} = R_{i,j,q} = 1$ if $\vec{\beta}_{[i:j]}$ contains only one non-zero entry.

Using the methods described in Section 2.3, 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.

2.3 INFERENCE, ESTIMATION, AND COMPUTATION

To extract useful inferences from the model of Section 2.2, we must address three sets of unknown quantities: the digestion template $\vec{t}$, the segmentation of each chromosome s, and the parameters and latent variables of the positioning model, $\vec{\beta}$, $\vec{\mu}$, and $\vec{\sigma}^2$. The parameter $\vec{\beta}$ and quantities derived from it are of the greatest scientific interest, as they correspond directly to the chromatin structure. However, before inferring $\vec{\beta}$, we address $\vec{t}$ and s.

To estimate the template t from paired-end sequencing data, we develop a nonparametric method, in Section 2.3.1. We develop a simple algorithm to segment each chromosome into non-overlapping segments with useful biological and statistical properties, in Section 2.3.2. Using the estimated template t and segmentation s, we turn to model-based inference for $(\vec{\beta}, \vec{\mu}, \vec{\sigma}^2)$. We build a parallel MCMC algorithm that can efficiently sample from the joint posterior of these parameters, in Section A.1.1. By combining the conditional independence structure of our model with distributed computation, we are able to handle datasets where $\vec{\beta}$ contains millions of entries.

An approximate EM algorithm is also provided in the online supplement 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 $(\vec{\beta}, \vec{\mu}, \vec{\sigma}^2)$ has a complex multimodal structure that EM is ill-equipped to address. Implementation details are given in the online supplement.

Finally, we calibrate the frequentist operating characteristics of our Bayesian estimators using a permutation null hypothesis, detailed in Section A.2.3. This ensures that our conclusions are valid both as Bayesian posterior probability assessments and under frequentist criteria. We focus on controlling the false discovery rate (FDR) for the detection of local structure in the distribution of nucleosome positions.

2.3.1 TEMPLATE ESTIMATION

Recall from Section 2.2.1 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. We show how $e_{1,j}$ and $e_{2,j}$ relate to each fragment in Figure 2.4. Along the genome, the distributions of digestion errors at the ends of each fragment are mirror images of each other, so positive values

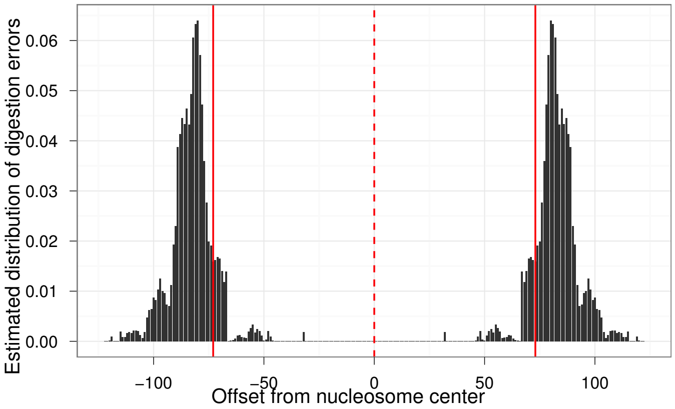


Figure 2.4: Estimated digestion error distributions vs. offset from nucleosome center; vertical lines at $\pm_0/2$ (solid) and nucleosome center (dashed).

imply that some DNA not bound to a nucleosome is under-digested, so $l_j > l_0$.

To setup our estimation problem, we define two probability distributions,

$$\begin{aligned} p(i) &= Pr\left(l_j = i\right) & (2.12) \\ q(i) &= Pr\left(e_{1,j} = i\right). & (2.13) \end{aligned}$$

Physically, $l_j \geq 0$, which implies $e_{1,j}, e_{2,j} \geq -\lfloor \frac{l_0}{2} \rfloor$. Analogously, if the longest observed fragment length is l_{max}, we have $Pr\left(l_j > l_{max}\right) = 0$. We require $l_j \leq l_{max}$, which imples $e_{1,j}, e_{2,j} \leq l_{max} - l_0 + \lfloor \frac{l_0}{2} \rfloor$. Thus, we can write

$$p(i) = \sum_{k=-\lfloor \frac{l_0}{2} \rfloor}^{l_{max}-l_0+\lfloor \frac{l_0}{2} \rfloor} q(k)q(i - l_0 - k). \qquad (2.14)$$

The resulting log-likelihood for the observed fragment lengths is $\sum_{j=1}^{M} \log p(l_j)$. We maximize this numerically, using a multivariate logit transformation on the values of $q(k)$ to avoid bounded optimization. Using the L-BFGS-B algorithm (Zhu et al., 1997) on a laptop with a Core i5 processor and 8GB of RAM, this maximization requires approximately 40 seconds for a typical experiment. This computation scales only with the number of unique fragment lengths observed, so it cannot become bottleneck for this method.

We obtain the template distribution t from q via a convolution sum and linear transformation. Recall from Section 2.2.1 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

$$u(i) = P(e_1 - e_2 = i) = \sum_{k=-\lfloor \frac{l_0}{2} \rfloor}^{l_{max}-l_0+\lfloor \frac{l_0}{2} \rfloor} q(k)q(k-i)\ . \tag{2.15}$$

We finally transform the distribution $u(i)$ to the desired template $t(i)$ by accounting for random rounding, as

$$t(k) = \frac{1}{2}u(2k-1) + u(2k) + \frac{1}{2}u(2k+1)\ . \tag{2.16}$$

Thus, the estimated template accurately reflects both the variation due to enzymatic digestion and the details of our preprocessing. We use this estimated template to build the design matrix X in the observation model, as discussed in Section 2.2, and for the simulation study discussed in Section 2.4.1.

2.3.2 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, non-overlapping 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 (800bp for the purposes of the analysis in Section 2.4).

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

$$c_i = \frac{1}{n_i} \sum_{k\,:\,s(k)=i} y_k\ , \tag{2.17}$$

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 $\min_i n_i < M$:
    Clear minimal difference in coverages $d_m$ and index $i_m$;
    /* Find the best merge among short segments                 */
    for $i : n_i < M$:
        Compute $d_i = \min(|c_i - c_{i-1}|, |c_i - c_{i+1}|)$;
        if $d_i < d_m$:
            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 $\{c_i\}$;
fintq
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 procedure maintains segment boundaries as a subset of ORF boundaries. This estimated segmentation is fixed and used in all subsequent inference.

2.3.3 POSTERIOR SAMPLER

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

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

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

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 $\vec{\beta}$ vectors are very high-dimensional. In some of the experiments discussed in Section 2.4, we work with simulated chromosomes with over 3.85 million base pairs. The conditional posterior of $\vec{\beta}^{(t)} | (\vec{\mu}^{(t)}, \vec{\sigma}^{2\,(t)})$ is not part of any standard family, so we turn to Hamiltonian Monte Carlo (HMC). The dimensionality of $\vec{\beta}$ makes a single HMC update for the entire vector both 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 $\vec{\beta}$ separated by at least $2w$ entries are conditionally independent given $(\vec{\mu}^{(t)}, \vec{\sigma}^{2\,(t)})$ and the entries of $\vec{\beta}$ between them. Consider the subvectors $\vec{\beta}_{[j_1:j_2]}$ and $\vec{\beta}_{[k_1:k_2]}$, with $j_1 < j_2 < k_1 < k_2$. The elements of $\vec{\beta}_{[j_1:j_2]}$ affect only $\vec{\lambda}_{[j_1-w:j_2+w]}$, and the elements of $\vec{\beta}_{[k_1:k_2]}$ affect only $\vec{\lambda}_{[k_1-w:k_2+w]}$. Hence, if $k_1 > j_2 + 2w$, then $\vec{\beta}_{[j_1:j_2]}$ and $\vec{\beta}_{[k_1:k_2]}$ are conditionally independent given $\vec{\mu}$ and $\vec{\sigma}^2$.

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 $\vec{\beta}$ into subvectors. The first starts at the beginning of $\vec{\beta}$ and proceeds forward with subvectors of length at most B separated by $2w$, yielding

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

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

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

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

Within each iteration of our sampler, we cycle through each of these partitions, updating each subvector of $\vec{\beta}$ with one HMC step. As each subvector within each parition 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. However, they are much faster than expected, 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 $\vec{t}$ instead, reducing the complexity of such products 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 supplement. A Python implementation of the sampler is available on GitHub, `www.github.com/awblocker/cplate`

2.3.4 Detection and calibration

Recall from Section 2.2.3 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 $P(C_{p,l}(k) > p/l \mid \vec{y})$ for each base pair k using the MCMC sampler described in Section 2.3. However, we require greater security in our detection results than the Bayesian approach alone can provide. To quantify the operating characteristics of our procedure and provide frequentist guarantees on its performance, we turn to a permutation null.

Our null hypothesis is that $\vec{y}$ consists of a set of multinomial draws. Under this null, the entries of $\vec{y}$ 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.

We then run our MCMC sampler on each such draw from the null, using the template and segmentation estimated from the observed data. From the sampler's output, we obtain an estimate of the distribution of $P(C_{p,l}(k) > p/l | \vec{y})$ 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 2.4, we have typically found that a threshold of approximately 0.8 on $P(C_{p,l}(k) > p/l|\vec{y})$ yields a FDR of 5% or less. This approach provides a secure detection procedure with both Bayesian and frequentist interpretations.

2.4 RESULTS

We demonstrate the proposed methods on real and simulated 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 2.4.4, 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., 2012).

The simulation studies aim to demonstrate the utility of the estimands introduced in Section 2.2.3 used in combination with the proposed deconvolution approach to inference, as well as the scalability of our methods. In Section 2.4.1, 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 2.4.2, we discuss the efficiency and scalability of inference via parallel Hamiltonian Markov Chain Monte Carlo sampling. In Section 2.4.3, 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., 2007a; 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 2.4.3, 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 2.4.3 and 2.4.3, 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. In Section 2.4.4, we assess the reproducibility of our inferences for cluster-level summaries of nucleosome positioning (Section 2.4.4) and the locations of local concentrations (Section 2.4.4). 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.

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, using data from chromosome I. For instance, sweeping τ_0 over two orders of magnitude (0.01–1) showed little effect on inferences, as did similar changes to (α_0, γ_0).

2.4.1 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 3501bp, consisting of a 1000bp promoter region, of a 2500bp coding region, and of a 1bp 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 to 95th percentile observed gene-level coverages in increments of 10%. Alternative position spacing had 10 levels, spanning from 0bp (no alternative positions) to 45bp in increments of 5bp. 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

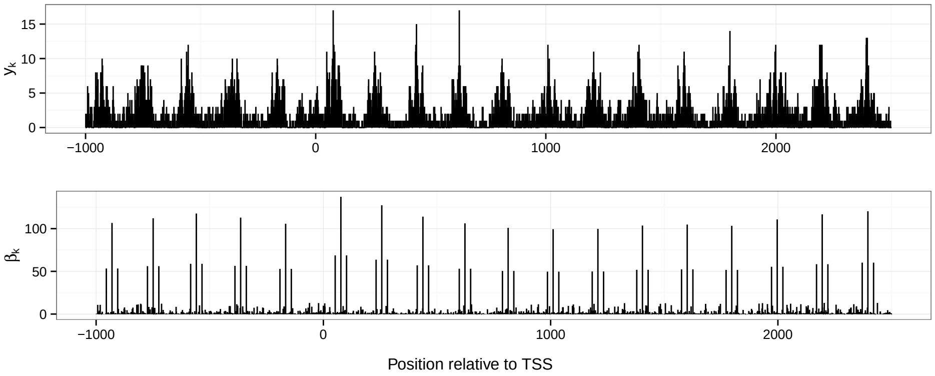


Figure 2.5: Illustration of one simulated gene: 0.55 quantile of coverage, with alternative position magnitude of 0.5, and alternative positions at ±25bp from each primary position. Read counts y_k (top panel), and coefficients β_k (bottom panel).

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 data sets, 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 1000bp before TSS to the end of each ORF, and computed the proportion of reads within the ORF observed within each such 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 2.5 provides an illustration of both coefficients, β_k, and read counts y_k, for one gene.

To generate our artificial dataset, we followed a modified version of the generative process outlined in Section 2.2. For each gene, we first drew coefficients for its subset of $\vec{\beta}$ 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 $\vec{\beta}$ 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 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 $\vec{\beta}$ vectors with the template estimated from the experimental data to obtain vectors of expected read counts $\vec{\lambda}$. Finally, we generated $\vec{y} \sim$ iid $\text{Poisson}(\vec{\lambda})$ to obtain simulated read counts. This entire procedure was repeated for each replicate, yielding 10 artificial chromosomes of length 3,851,100bp each.

These simulations are inspired by our generative model, but they do not follow it's structure to the letter. The distribution of background coefficients and the effects of digestion agree between our model and our simulations. However, we introduce much more structure into the locations of nucleosome concentration via the deterministic placement of primary and alternative positions. This simultaneously provides a stringent test of our methods and decreases the amount of residual variation across our experimental replicates. 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 supplement.

2.4.2 Parallel HMC performance

The parallel Hamiltonian Monte Carlo sampler performed well on both real and simulated datasets, based on standard MCMC diagnostics. For the actual and simulated datasets used in Section 2.4, we ran 2,000 iterations, discarding the first 200 as burn-in. This yielded 1,800 draws for $\vec{\beta}$, $\vec{\mu}$, and $\vec{\sigma}^2$. The mean effective sample size for the elements of $\vec{\theta} = \log \vec{\beta}$ 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 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 extremely scalable. Using 144 cores on the Harvard Odyssey cluster and setting $B = 2000$, each simulated chromosome required 1.83

seconds per iteration for a total runtime of approximately 1 hour. The smallest S. cerevisiae chromosome (I) required 0.136 seconds per iteration, while the largest (IV) required 0.699 second per iteration, yielding total runtimes of 4.5 and 23.3 minutes, respectively. Running the entire S. cerevisiae genome required approximately 3.24 hours. The sampler was also run on an Amazon EC2 cluster with 512 cores, processing the same genome in under an hour.

2.4.3 POWER ANALYSIS

Using simulated chromosomes, we compare the performance of the proposed method to that of a Parzen-window estimator for estimating locations of clusters of nucleosome positions, and assess the performance of the proposed methods for detecting and estimating the locations of both 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 2.4.1, along with their coefficients. Recall that the output of the calibrated detection procedure, detailed in Section A.2.3, 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 all ground truth positions, primary and alternative. Finding the nearest estimated position for each ground truth position yields measures of power, as we can measure the distance from each true position to the nearest inferred one. Large distances imply low sensitivity, and vice versa. Conversely, finding the nearest ground truth position for each inferred position yields measures of accuracy. If inferred positions are far from the true ones, we would consider the results unreliable.

The analyses below are based on summaries of these matched distances; we compute mean and median absolute errors, and we tabulate the proportion of

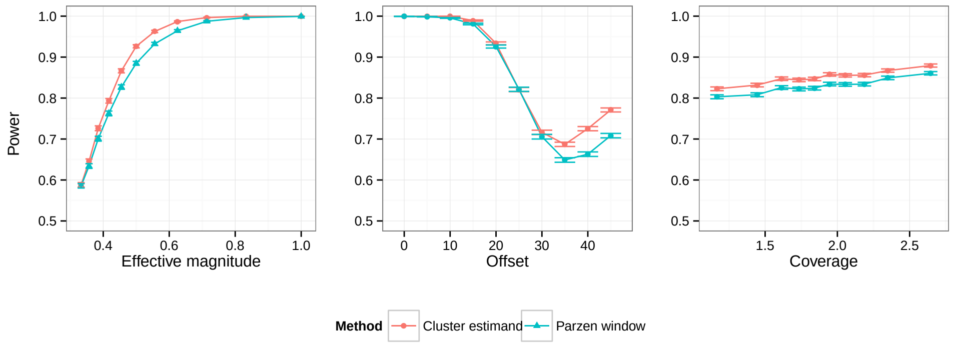


Figure 2.6: Power vs. effective magnitude (left), alternative position offset (center), and coverage (right) for Parzen window and cluster estimand methods

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 directly interpretable as a measure of power.

CLUSTERS

Detection of a cluster was defined as a best-match distance of less than 5bp between the inferred and true cluster center (κ_m). Figure 2.6 summarizes our key findings, showing the relative power of each method against the effective magnitude of the primary position, the offset of the alternative positions, and gene-level coverage. Tables 2.1 provides the results of a design-based ANOVA of the mean absolute errors of estimated cluster locations, by gene, and Table 2.2 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 both in terms of 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 to 60bp. 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

		Cluster Estimand		Parzen Window	
	Df	Sum Sq	F value	Sum Sq	F value
Coverage	9	1928.94	198.73	753.24	79.04
Offset	9	19422.58	2001.01	10789.06	1132.10
Magnitude	9	10764.55	1109.02	6768.51	710.22
Coverage:Offset	81	275.41	3.15	264.68	3.09
Coverage:Magnitude	81	175.19	2.01	142.80	1.66
Offset:Magnitude	72	23535.40	303.09	19172.33	251.47
Coverage:Offset:Magnitude	648	950.25	1.36	947.80	1.38
Residuals	10090	10881.93		10684.31	

Table 2.1: Analysis of variance of absolute errors in cluster centers for cluster estimand and Parzen window methods. All factors and interactions were statistically significant with $p < 0.0001$.

	Cluster Estimand		Parzen Window	
	Estimate	z value	Estimate	z value
(Intercept)	3.3461	54.15	3.4660	56.74
Coverage	0.6069	5.20	0.7498	6.54
Offset	-5.3115	-58.00	-5.4596	-60.92
Effective Magnitude	2.7211	10.76	2.3296	10.82
Coverage·Offset	-0.4217	-2.48	-0.6071	-3.66
Coverage·Effective Magnitude	1.7927	3.20	0.9037	2.02
Offset·Effective Magnitude	8.9842	22.27	6.9932	21.15
Coverage·Offset·Effective Magnitude	2.1476	2.52	1.8962	2.84

Table 2.2: Logistic regression of power on design factors as continuous variables cluster estimand and Parzen Window method. Are regressors are normalized to have range $[0, 1]$.

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 to 35bp. Power increases slightly for both methods at offsets of 40 and 45bp. The relative performance of the proposed method is largest for offsets over 30bp, with a difference in power of 7% at 45bp. Power shows little marginal dependence upon local coverage with only a 6% change in power over the range of coverages for both methods.

The ANOVA and logistic regression analyses support these observations and provide further insights into the role of interactions between the design factors. ANOVA results in Table 2.1 indicate that alternative position offset, effective magnitude, and their interaction account for the vast 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 2.2 suggest that 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 positve, 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.

Taken together, these results demonstrate that the proposed method offers improved performance relative to the standard method in the field, for estimating cluster locations. However, the proposed method offers the greatest benefits for exploring local concentration in the distribution of nucleosome positions, an estimand that the Parzen-window estimator cannot reliably infer.

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 2.8. For this analysis, we fix $l = 73$ and $p = 3$, and used the calibrated detection procedure described in Section A.2.3 with a maximum FDR of 5%.[1] For primary positions, we declare a successful detection if the best-match distance is less than 5bp 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 position and the true alternative position is less than 1/2 of the alternative position's distance from its primary position. Figure 2.7 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.

---

[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.

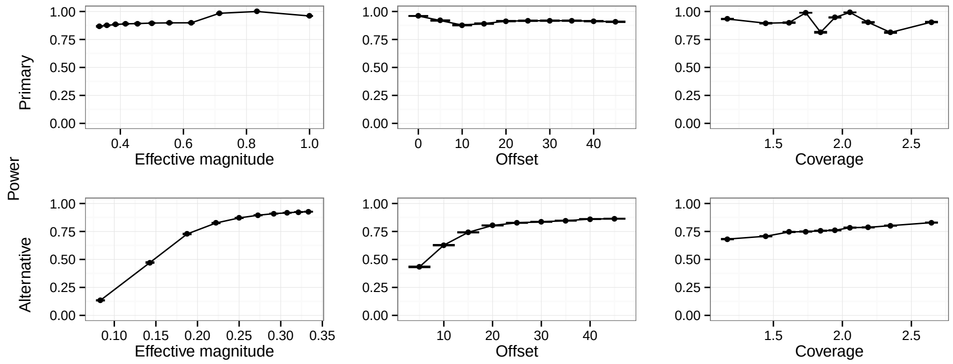


Figure 2.7: Power vs. effective magnitude (left), alternative position offset (center), and coverage (right) for detection of primary and alternative positions ±3bp

	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	

Table 2.3: Analysis of variance of absolute position errors for the detection of primary and alternative positions using local concentration estimands. * indicates that a factor was statistically significant with $p < 0.0001$. Remaining factors had p-values larger than 0.95.

Tables 2.3 provides the results of a design-based ANOVA of the mean absolute errors of estimated cluster locations, by gene, and Table 2.4 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 to 6.61bp and from 2.17 to 41.8bp, respectively. The sensitivity of power and absolute position errors

	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

Table 2.4: Logistic regression of power on design factors as continuous variables for primary and alternative positions. Are regressors are normalized to have range $[0,1]$.

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 2.3 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 2.4 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 2.3 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 2.4 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 2.1bp. 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 6.0bp. We discuss the implications of these capabilities for biological analyses in Section 2.5.

2.4.4 REPRODUCIBILITY ANALYSIS

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. 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 data sets, H_{ij} for $i, j = 1, 2$, which allow for two comparisons within biological replicates (i.e., H_{11} versus H_{12}, and H_{21} versus 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 2.2.1.

We examine the reproducibility of inferences on cluster-level and local concentration estimands in Sections 2.4.4 and 2.4.4, 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 both cluster-level properties and local concentrations. We present detailed results for each class of estimand below.

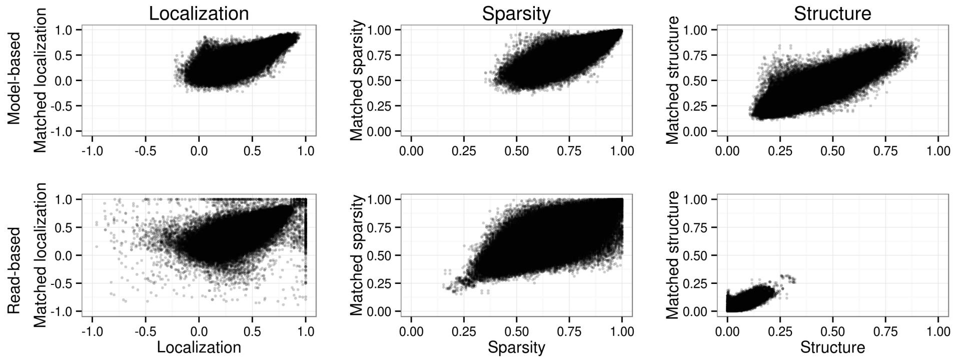


Figure 2.8: Joint distributions of local, structure, and sparsity indices for matched clusters between biological replicates for model-based (top) and Parzen window/read-based (bottom).

CLUSTERS

We assessed the reproducibility of estimated cluster positions and cluster-level summaries from both our method and the standard Parzen-window technique. For the former, all estimates are posterior means of the estimands specified in Equations 2.9–2.11 ($L_{i,j}$, $S_{i,j}$, and $R_{i,j,q}$) using a window of ±73bp 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 $\vec{y}$ directly to obtain estimates of the localization, sparsity, and structure indices described in Section 2.2.3 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 2.8.

Inferred cluster positions were highly reproducible with a mean best-match distance of 15.72±0.14bp and median best-match distance of 4bp, between biological replicates, and of 14.30±0.2bp and 3bp, respectively, between technical replicates. With the proposed method, 90% of clusters were matched within 44bp across biological replicates, and within 35bp 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 ± 0.14bp and 4bp, between biological replicates, and of 13.98 ± 0.19bp and 8bp, 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 ± 0.07bp and 1bp. Across methods, 90% of inferred cluster positions were matched within 2bp and 95% were matched within 3bp.

Cluster-level properties, however, showed significant differences between the model-based and Parzen-window estimates, both in terms of reproducibility and comparability, as Figure 2.8 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 which 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. However, 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 2.8: the read-based localization index is noisier than the model-based one, but their distributions appear otherwise comparable. 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 than the correlation suggests, with the model-based estimator spanning nearly 3 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 2.8. 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

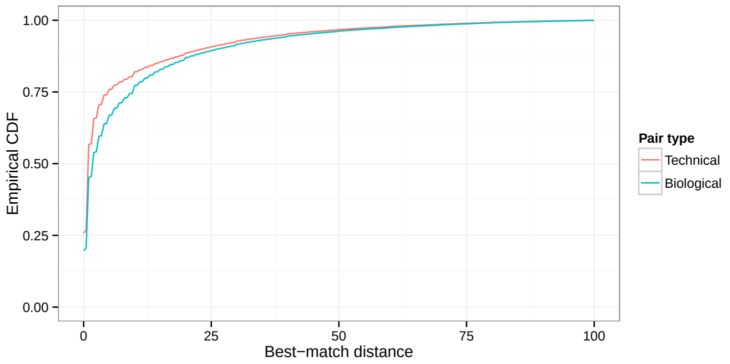


Figure 2.9: Empirical CDFs of best-match distances between detected local concentrations for technical (red) and biological (blue) replicates.

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 methods provide reproducible inferences about the local structure of nucleosome positions across variation from biological and technical sources, including explicit changes to the degree of enzymatic digestion. They significantly outperform standard Parzen-window and read-based estimators in this regard and provide a richer, more accurate view of the true distribution of nucleosome positions.

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 2.9, where we compare the distributions of best-match distances between biological and technical replicates.

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

2.5 CONCLUDING REMARKS

We have presented an approach to modeling and making inferences about the genome-wide distribution of nucleosome positions from paired-end sequencing data. The results presented in Section 2.4 demonstrate the utility of the proposed methods for biological analyses, particularly the reproducibility of its inferences across experimental conditions. 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 the 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 data.

2.5.1 MODELING

We explicitly choose not to include prior information on nucleosome spacing in this model. Previous work has used the empirically-observed 150-200bp spacing between nucleosomes within individual cells to constrain inferences on nucleosome positions (e.g., see Shivaswamy et al., 2008; Yuan et al., 2005). In the presence of alternative nucleosome positioning and chromatin dynamics, however, constraints on spacing that hold on a single-cell level need not hold after aggregation across a population of cells, which is where measurements are taken. 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. Thus, the proposed model does not use information on expected separation among nucleosomes along the sequence to constrain the inferred nucleosome positions. 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 by pairs of reads that is provided by paired-end sequencing technology to infer the effects of enzymatic digestion on the measurements $\vec{y}$. Many studies, however, use single-end sequencing technology, which does not provide fragment lengths. In related work (Zhou et al., 2012), we have developed an approach to estimate the digestion variability template, $\vec{t}$, for single-end data using an alternative source of fragment-length information; Bioanalyzer technology (e.g., Mueller, 2000). The model and inference presented in Sections 2.2 and 2.3 can be adapted to this single-end context with an appropriate modification of the template $\vec{t}$ and of the digestion matrix X.

2.5.2 ESTIMANDS

In defining the estimands of biological interest, we aimed at separating 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 2.2, the distribution of nucleosome positions corresponds to the $\vec{\beta}$ vector, while the template $\vec{t}$ and the remaining error structure capture other sources of experimental variation. The estimands presented in Section 2.2.3 are functions only of the true underlying $\vec{\beta}$, 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 cluster-based estimands such as $L_{i,j}$, $S_{i,j}$, and $R_{i,j,q}$ and other summaries of local structure, such as those based on $C_{p,l}(k)$. Cluster-based estimands capture the properties of the distribution of nucleosome positions within small regions identified by a clustering algorithm. These measured depend on the particular definition of "cluster" and on the clustering method used. The sensitivity of these estimands to these choices is problematic, in practice, and fine-grained structure is lost in the reduction of data to clusters.

However, this reduction can simplify subsequent interpretation. In contrast, estimands such as the local concentration index $C_{p,l}(k)$ summarize the local structure of the nucleosome position distribution without relying on a clustering criterion. They lead to reproducible analyses and can be relied upon for scientific discovery of small-scale features. We believe that these classes of estimands are most useful in combination, providing complementary views on the distribution of nucleosome positions.

Second, we have found that the magnitude of the performance gains stemming from model-based inferences depends strongly on the estimand of interest. For instance, the proposed method outperforms read-based estimators in terms of power and error when targeting our cluster-level localization estimands L, but the difference in reproducibility is not overwhelming. However, the increase in reproducibility one can expect from the proposed method is substantial when targeting structure and sparsity measures, S and R. This reflects the greater sensitivity of read-based estimators of structure and sparsity to observation noise. In addition, as we have shown in Section 2.4.3 and 2.4.4, the proposed method can provide reproducible inferences about individual local features less than 10bp wide, when inference on properties of the distribution of nucleosome positions at such a fine resolution has been so far considered infeasible with read-based estimators. More generally, the more sensitive an estimand is to observation noise, the greater the performance gains expected from using the proposed method.

Our results suggest that careful probabilistic modeling of the core sources of experimental variation can enable new types of scientific inferences.

2.5.3 INFERENCE

The use of distributed computing was essential for our method, as it allowed us to sample from the marginal posterior of $\vec{\beta}$ in only minutes per chromosome. We leveraged the conditional independence structure of our model to create an efficient, scalable distributed MCMC sampler. This structure stems from the finite length of the digestion variability template $\vec{t}$. As the template is $2w + 1$ wide, subvectors of $\vec{\beta}$ separated by at least $2w$ base pairs are conditionally independent *a posteriori* given $(\vec{\mu}, \vec{\sigma}^2)$. 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 $(\vec{\mu}, \vec{\sigma}^2)$ are needed for each update.

To update each subvector of $\vec{\beta}$ in our MCMC, 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 $\vec{\beta}$ 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 2.4.2, 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 define the structure of interest. We then use draws from the parallel HMC sampler to estimate the posterior probabilities of these estimands exceeding 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). Instead of relying upon the model to provide a null distribution for such calibration, 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 our Bayesian posterior probabilities to the more standard, interpretable scale of FDRs and q-values.

The pragmatic approach to detection described above is one step in our broader approach to the analysis of nucleosome positioning. First, we used a probability model to build statistics that directly target the scientific estimands of interest, and performed inference with MCMC. Second, we used permutations to define a reference distribution based on the observed data and the segmentation, and detect local concentrations of nucleosome positions. Third, we evaluated the power, accuracy, and reproducibility of inferences from our method using biologically-motivated simulations, and technical and biological replicates. Each step in this 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 Section 2.4. 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.

Bibliography

Istvan Albert, Travis N Mavrich, Lynn P Tomsho, Ji Qi, Sara J Zanton, Stephan C Schuster, and B Franklin Pugh. Translational and rotational settings of H2A.Z nucleosomes across the Saccharomyces cerevisiae genome. *Nature*, 446(7135): 572–6, March 2007a. ISSN 1476-4687. doi: 10.1038/nature05632. URL `http://www.ncbi.nlm.nih.gov/pubmed/17392789`.

Istvan Albert, Travis N Mavrich, Lynn P Tomsho, Ji Qi, Sara J Zanton, Stephan C Schuster, and B Franklin Pugh. Translational and rotational settings of H2A.Z nucleosomes across the Saccharomyces cerevisiae genome. *Nature*, 446(7135): 572–6, March 2007b. ISSN 1476-4687. doi: 10.1038/nature05632. URL `http://www.ncbi.nlm.nih.gov/pubmed/17392789`.

A. Barski and K. Zhao. Genomic location analysis by ChIP-Seq. *Journal of Cellular Biochemistry*, 107:11–18, 2009.

Kristin Brogaard, Liqun Xi, Ji-Ping Wang, and Jonathan Widom. A map of nucleosome positions in yeast at base-pair resolution. *Nature*, 486 (7404):496–501, June 2012. ISSN 1476-4687. doi: 10.1038/nature11142. URL `http://www.nature.com/doifinder/10.1038/nature11142http://www.ncbi.nlm.nih.gov/pubmed/22722846`.

Jonathan Cairns, Christiana Spyrou, Rory Stark, Mike L Smith, Andy G Lynch, and Simon Tavaré. BayesPeak–an R package for analysing ChIP-seq data. *Bioinformatics (Oxford, England)*, 27(5):713–4, March 2011. ISSN 1367-4811. doi: 10.1093/bioinformatics/btq685. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3042177&tool=pmcentrez&rendertype=abstract`.

Oscar Flores and Modesto Orozco. nucleR: a package for non-parametric nucleosome positioning. *Bioinformatics (Oxford, England)*, 27(15):2149–50, August 2011. ISSN 1367-4811. doi: 10.1093/bioinformatics/btr345. URL `http://www.ncbi.nlm.nih.gov/pubmed/21653521`.

Kai Fu, Qianzi Tang, Jianxing Feng, X Shirley Liu, and Yong Zhang. DiNuP: a systematic approach to identify regions of differential nucleosome positioning. *Bioinformatics (Oxford, England)*, 28(15):1965–71, August 2012. ISSN 1367-4811. doi: 10.1093/bioinformatics/bts329. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3400967&tool=pmcentrez&rendertype=abstract`.

T. Gkikopoulos, P. Schofield, V. Singh, M. Pinskaya, J. Mellor, M. Smolle, J. L. Workman, G. J. Barton, and T. Owen-Hughes. A Role for Snf2-Related Nucleosome-Spacing Enzymes in Genome-Wide Nucleosome Organization. *Science*, 333(6050):1758–1760, September 2011. ISSN 0036-8075. doi: 10.1126/science.1206097. URL `http://www.sciencemag.org/cgi/doi/10.1126/science.1206097`.

Mayetri Gupta. Generalized hierarchical markov models for the discovery of length-constrained sequence features from genome tiling arrays. *Biometrics*, 63(3):797–805, September 2007. ISSN 0006-341X. doi: 10.1111/j.1541-0420.2007.00760.x. URL `http://www.ncbi.nlm.nih.gov/pubmed/17825011`.

An Jansen and Kevin J Verstrepen. Nucleosome Positioning in Saccharomyces cerevisiae. *Microbiology and molecular biology reviews : MMBR*, 75(2):301–20, June 2011. ISSN 1098-5557. doi: 10.1128/MMBR.00046-10. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3122627&tool=pmcentrez&rendertype=abstract`.

B. Langmead, C. Trapnell, M. Pop, and S. L. Salzberg. Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biology*, 10(3):R25, 2009.

William Lee, Desiree Tillo, Nicolas Bray, Randall H Morse, Ronald W Davis, Timothy R Hughes, and Corey Nislow. A high-resolution atlas of nucleosome occupancy in yeast. *Nature genetics*, 39(10):1235–44, October 2007. ISSN 1546-

1718. doi: 10.1038/ng2117. URL `http://www.ncbi.nlm.nih.gov/pubmed/17873876`.

Ritendranath Mitra and Mayetri Gupta. A continuous-index Bayesian hidden Markov model for prediction of nucleosome positioning in genomic DNA. *Biostatistics (Oxford, England)*, 12(3):462–77, July 2011. ISSN 1468-4357. doi: 10.1093/biostatistics/kxq077. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3114652&tool=pmcentrez&rendertype=abstract`.

Odilo Mueller. High precision restriction fragment sizing with the Agilent 2100 Bioanalyzer Application Note. 2000.

P. J. Park. ChIP-Seq: Advantages and challenges of a maturing technology. *Nature Reviews Genetics Genet.*, 10:669–680, 2009.

Shirley Pepke, Barbara Wold, and Ali Mortazavi. Computation for ChIP-seq and RNA-seq studies. *Nature methods*, 6(11 Suppl):S22–32, November 2009. ISSN 1548-7105. doi: 10.1038/nmeth.1371. URL `http://www.ncbi.nlm.nih.gov/pubmed/19844228`.

Anton Polishko, Nadia Ponts, Karine G Le Roch, and Stefano Lonardi. NORMAL: accurate nucleosome positioning using a modified Gaussian mixture model. *Bioinformatics (Oxford, England)*, 28(12):242–9, June 2012. ISSN 1367-4811. doi: 10.1093/bioinformatics/bts206. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3371838&tool=pmcentrez&rendertype=abstract`.

Naim U Rashid, Paul G Giresi, Joseph G Ibrahim, Wei Sun, and Jason D Lieb. ZINBA integrates local covariates with DNA-seq data to identify broad and narrow regions of enrichment, even within amplified genomic regions. *Genome biology*, 12(7):R67, January 2011. ISSN 1465-6914. doi: 10.1186/gb-2011-12-7-r67. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3218829&tool=pmcentrez&rendertype=abstract`.

Armin Schwartzman, Andrew Jaffe, Yulia Gavrilov, and Clifford A Meyer. Multiple testing of local maxima for detection of peaks in ChIP-Seq data. 2011.

E. Segal, Y. Fondufe-Mittendorf, L. Chen, A. Thastrom, Y. Field, I. K. Moore, J. Z. Wang, and J. Widom. A genomic code for nucleosome positioning. *Nature*, 42: 772–778, 2006.

Sushma Shivaswamy, Akshay Bhinge, Yongjun Zhao, Steven Jones, Martin Hirst, and Vishwanath R Iyer. Dynamic remodeling of individual nucleosomes across a eukaryotic genome in response to transcriptional perturbation. *PLoS biology*, 6(3):e65, March 2008. ISSN 1545-7885. doi: 10.1371/journal.pbio.0060065. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2267817&tool=pmcentrez&rendertype=abstract`.

J.D. Storey and R. Tibshirani. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences of the United States of America*, 100 (16):9440, 2003.

Wei Sun, Michael J Buck, Mukund Patel, and Ian J Davis. Improved ChIP-chip analysis by a mixture model approach. *BMC bioinformatics*, 10:173, January 2009a. ISSN 1471-2105. doi: 10.1186/1471-2105-10-173. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2700807&tool=pmcentrez&rendertype=abstract`.

Wei Sun, Wei Xie, Feng Xu, Michael Grunstein, and Ker-Chau Li. Dissecting nucleosome free regions by a segmental semi-Markov model. *PloS one*, 4(3):e4721, January 2009b. ISSN 1932-6203. doi: 10.1371/journal.pone.0004721. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2648986&tool=pmcentrez&rendertype=abstract`.

Itay Tirosh. Computational analysis of nucleosome positioning. *Methods in molecular biology (Clifton, N.J.)*, 833:443–9, January 2012. ISSN 1940-6029. doi: 10.1007/978-1-61779-477-3_27. URL `http://www.springerlink.com/index/10.1007/978-1-61779-477-3http://www.ncbi.nlm.nih.gov/pubmed/22183610`.

Alexander M Tsankov, Dawn Anne Thompson, Amanda Socha, Aviv Regev, and Oliver J Rando. The role of nucleosome positioning in the evolution of gene regulation. *PLoS biology*, 8(7):e1000414, January 2010. ISSN 1545-7885. doi: 10.1371/journal.pbio.1000414. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2897762&tool=pmcentrez&rendertype=abstract.

Assaf Weiner, Amanda Hughes, Moran Yassour, Oliver J Rando, and Nir Friedman. High-resolution nucleosome mapping reveals transcription-dependent promoter packaging. *Genome research*, 20(1):90–100, January 2010. ISSN 1549-5469. doi: 10.1101/gr.098509.109. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2798834&tool=pmcentrez&rendertype=abstract.

M. Yassour, T. Kaplan, A. Jaimovich, and N. Friedman. Nucleosome positioning from tiling microarray data. *Bioinformatics 24*, 24:i139–146, 2008.

Guo-Cheng Yuan and Jun S Liu. Genomic sequence is highly predictive of local nucleosome depletion. *PLoS computational biology*, 4(1): e13, January 2008. ISSN 1553-7358. doi: 10.1371/journal.pcbi.0040013. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2211532&tool=pmcentrez&rendertype=abstract.

Guo-Cheng Yuan, Yuen-Jong Liu, Michael F Dion, Michael D Slack, Lani F Wu, Steven J Altschuler, and Oliver J Rando. Genome-scale identification of nucleosome positions in S. cerevisiae. *Science*, 309(5734):626–30, July 2005. ISSN 1095-9203. doi: 10.1126/science.1112178. URL http://www.ncbi.nlm.nih.gov/pubmed/15961632.

Xuekui Zhang, Gordon Robertson, Sangsoon Woo, Brad G Hoffman, and Raphael Gottardo. Probabilistic inference for nucleosome positioning with MNase-based or sonicated short-read data. *PloS one*, 7(2):e32095, January 2012. ISSN 1932-6203. doi: 10.1371/journal.pone.0032095. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3290535&tool=pmcentrez&rendertype=abstract.

Yong Zhang, Tao Liu, Clifford a Meyer, Jérôme Eeckhoute, David S Johnson, Bradley E Bernstein, Chad Nusbaum, Richard M Myers, Myles Brown, Wei Li, and X Shirley Liu. Model-based analysis of ChIP-Seq (MACS). *Genome biology*, 9(9):R137, January 2008. ISSN 1465-6914. doi: 10.1186/gb-2008-9-9-r137.

URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2592715&tool=pmcentrez&rendertype=abstract`.

X. Zhou, A. W. Blocker, E. M. Airoldi, and E. K. O'Shea. A genome-wide analysis of chromatine remodeling after phosphate starvation. Manuscript, 2012.

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`.

A

Online supplement for “Template-based estimation of genome-wide nucleosome positioning via distributed HMC”

A.1 ALGORITHMIC DETAILS OF INFERENCE

A.1.1 DISTRIBUTED HMC SAMPLER

Recall that the model specified in Section 2 is:

$$
\begin{aligned}
y_k | \lambda_k &\sim \text{Poisson}\left(\lambda_k\right) && \text{(A.1)} \\
\lambda_{(N \times 1)} &\equiv X_{(N \times (N - \ell_0))}\, \beta_{((N - \ell_0) \times 1)}, && \text{(A.2)} \\
& \qquad \beta_k > 0 \text{ for } k = \lfloor \ell_0 / 2 \rfloor + 1 \ldots N - \lfloor \ell_0 / 2 \rfloor \\
\log \beta_k &\sim \text{Normal}\left(\mu_{s_k}, \sigma^2_{s_k}\right) && \text{(A.3)}
\end{aligned}
$$

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_s^2):

$$\sigma_s^2 \sim \text{InvGamma}(\alpha_0, \gamma_0) \ , \tag{A.4}$$

$$\mu_s | \sigma_s^2 \sim N(\mu_0, \frac{\sigma_s^2}{n_s \tau_0}) \ , \tag{A.5}$$

where n_s is the length of segment s.

Our MCMC sampler alternates between two conditional updates:

1. Draw $(\vec{\mu}^{(r)}, \vec{\sigma}^{2\,(r)}) | \vec{\beta}^{(r-1)}$ directly, then
2. Update $\vec{\beta}^{(r)} | (\vec{\mu}^{(r)}, \vec{\sigma}^{2\,(r)})$ 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.

HYPERPARAMETER UPDATES

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

$$\begin{aligned} \sigma_s^{2\,(r)} | \vec{\beta}^{(r-1)} &\sim \mathit{InvGamma}\Bigg(\frac{n_s}{2} + \alpha_0, \ \frac{1}{2} \sum_{k:s_k=s} (\theta_k^{(r-1)} - \bar{\theta}_s^{(r-1)})^2 \qquad \text{(A.6)} \\ &\quad + \frac{\tau_0 n_s}{2(1+\tau_0)} (\bar{\theta}_s^{(r-1)} - \mu_0)^2 + \gamma_0 \Bigg) \ , \\ \mu_s^{(r)} | \sigma_s^{2\,(r)}, \vec{\beta}^{(r-1)} &\sim N\left(\frac{\bar{\theta}_s^{(r-1)} + \tau_0 \mu_0}{1+\tau_0}, \frac{\sigma_s^{2\,(r)}}{n_s(1+\tau_0)} \right) \ , \qquad \text{(A.7)} \end{aligned}$$

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 $O(N)$.

DISTRIBUTED HMC UPDATE FOR $\vec{\beta}$

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

separated by $2w$, yielding

$$D_1 = \vec{\beta}_{[1:B]}, \vec{\beta}_{[B+2w+1:2B+2w]}, \ldots, \vec{\beta}_{[n_b(B+2w)+1:N]}, \tag{A.8}$$

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

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

```
Distributed HMC update
 /* Send conditioning information                                  */
 Broadcast $\vec{\mu}^{(t-1)}$, and $\vec{\sigma}^{2\,(t-1)}$ 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 $\vec{\theta}[start : end]$ 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 $\vec{\theta}[start : end]$ from arbitrary worker with tag $b_1$ ;
         Incorporate result into working copy of $\vec{\theta}^{(t)}$ ;
         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 $\vec{\theta}[start : end]$ to last completed worker process with work tag
             attached;
             nStarted++;
     fintq
```

Algorithm 2: Distributed HMC update

The individual, worker-level HMC updates are done on $\vec{\theta} = \log \vec{\beta}$ and follow the standard leapfrog-based HMC procedure outlined in **?**. To compute these HMC updates, we require the log-posterior density of each subvector of $\vec{\theta}$ and its gradient. First, define $\vec{\lambda} = X\vec{\beta}$, $\vec{m} = (\mu_{s_k} \text{ for } k = 1, \ldots, N)^\top$, and $\vec{v} = (\sigma^2_{s_k} \text{ for } k = 1, \ldots, N)^\top$. Also, for vectors of equal dimension, let $/$ denote entrywise division and $**$ denote entrywise powers. Then,

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

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

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

where $W = \text{diag}\left(\vec{y}/\vec{\lambda}^{**}2\right)$. 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 $\vec{t}$. This also enables efficient computation of the Hessian's diagonal, as

$$\vec{\lambda} = X\vec{\beta} = (\vec{0}^\top_{\lfloor \ell_0/2 \rfloor}\ \vec{\beta}^\top\ \vec{0}^\top_{\lfloor \ell_0/2 \rfloor})^\top * \vec{t} \tag{A.13}$$

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

$$\text{diag}\left(X' W X\right) = (\vec{y}/\vec{\lambda}^{**}2) * \vec{t}^{**}2\,. \tag{A.15}$$

This reduces the computational complexity of these evaluations to $O(B \log B)$ for each update of each subvector of $\vec{\beta}$. Our block-level HMC steps are detailed in Algorithm 3. In practice, we fix $\epsilon_{\text{min}} = 0.001$, $\epsilon_{\text{max}} = 0.1$, 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

```
Data: Trajectory length $L$, $\vec{\mu}$, $\vec{\sigma}^2$, $\vec{\theta}[start : end]$, $\epsilon$min, $\epsilon_{\max}$, block start $b$,
      template $\vec{t}$, chromosome length $N$
/* Subset $\vec{\theta}[start : end]$ to B-length subvector to update and
   buffers                                                          */
$\tilde{\vec{\theta}} = \vec{\theta}[b : \min(b + B - 1, N)]$;
$\tilde{\vec{\theta}}_0 = \tilde{\vec{\theta}}; \underline{\vec{\theta}} = (\vec{\theta}[start : b], \vec{\theta}[\min(b + B - 1, N) : end])$;
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 $\tilde{\vec{\theta}}$;
    $M = -\nabla_{\tilde{\vec{\theta}}} \nabla_{\tilde{\vec{\theta}}^\top} \log p(\tilde{\vec{\theta}} | \vec{\mu}, \vec{\sigma}^2, \underline{\vec{\theta}})$;
    if Using diagonal mass matrix:
        $M = \text{diag}(M)$;
else:
    $M = I_{end-start}$;
/* Draw momentum                                                    */
Draw $\vec{p} \sim N(\vec{0}, M)$;
$\vec{p}_0 = \vec{p}$;
/* Run leapfrog integration                                         */
$\vec{p} += \epsilon \nabla_{\tilde{\vec{\theta}}} \log p(\tilde{\vec{\theta}} | \vec{\mu}, \vec{\sigma}^2, \underline{\vec{\theta}})/2$;
for $i$ in range($L$):
    $\tilde{\vec{\theta}} += \epsilon M^{-1} p$;
    if $i < L - 1$:
        $\vec{p} += \epsilon \nabla_{\tilde{\vec{\theta}}} \log p(\tilde{\vec{\theta}} | \vec{\mu}, \vec{\sigma}^2, \underline{\vec{\theta}})$;
$\vec{p} += \epsilon \nabla_{\tilde{\vec{\theta}}} \log p(\tilde{\vec{\theta}} | \vec{\mu}, \vec{\sigma}^2, \underline{\vec{\theta}})/2$;
/* Metropolis-Hastings step to correct for integration
   errors                                                           */
$\log r = \log p(\tilde{\vec{\theta}} | \vec{\mu}, \vec{\sigma}^2, \underline{\vec{\theta}}) - \log p(\tilde{\vec{\theta}}_0 | \vec{\mu}, \vec{\sigma}^2, \underline{\vec{\theta}}) - 1/2(\vec{p}^\top M^{-1} \vec{p} - \vec{p}_0^\top M^{-1} \vec{p}_0)$;
Draw $u$ Unif$[0, 1]$;
if $u \leq r$:
    return $(\tilde{\vec{\theta}}, 1)$ ;                      /* Accept update */
else:
    return $(\tilde{\vec{\theta}}_0, 0)$ ;                    /* Reject update */
```

Algorithm 3: Worker-level HMC update

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.1.2 APPROXIMATE EM ALGORITHM

We develop an approximate EM algorithm, based on a Gaussian approximation of the conditional posterior of $\vec{\theta}$, as to obtain starting values for the MCMC sampler given in A.1.1. It provides a high-quality initialization for $\vec{\beta}$, $\vec{\mu}$, and $\vec{\sigma}^2$. Simpler initializations are possible, but obtaining high-quality initial estimates can greatly reduce the number of MCMC iterations required for reliable inferences.

CHOICE INITIAL ESTIMATOR

We use $\hat{\theta}_k = E\left[\theta_k | \vec{y}, \hat{\mu}_{s_k}, \hat{\sigma}^2_{s_k}\right]$ as an initial point estimate of θ_k. The distributed HMC sampler presented in Section A.1.1 yields information on the complete marginal posterior of $\vec{\theta}$ 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.1.2 provides both approximate marginal MAP estimates of the parameters $\vec{\mu}$ and $\vec{\sigma}^2$ and estimates of the target conditional expectations $\hat{\theta}_k$.

APPROXIMATE EM ALGORITHM VIA LAPLACE APPROXIMATION

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

APPROXIMATE E-STEP In the E-step, the objective is to compute

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

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

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

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

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

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

While the conditional posterior $p(\vec{\theta} | \vec{y}, \vec{\mu}, \vec{\sigma}^2)$ 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 $\{\mu_{s_k}, \sigma_{s_k}^2\}$ 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 $(\vec{\mu}^{(r-1)}, \vec{\sigma}^{2\,(r-1)})$. This amounts to maximizing

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

with respect to $\vec{\theta}$. 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 $\vec{\theta}_t$ equal to the mode of g, and variance $\hat{V}_k = -\text{diag}\,(H^{-1})_k$ for the conditional posterior $p(\theta_k|\vec{y}, \mu_{s_k}, \sigma^2_{s_k})$.

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

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

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

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 $\vec{\sigma}^2 = \vec{0}$ 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.

DISTRIBUTED APPROXIMATE E-STEP

We use the conditional independence structure of $\vec{\beta}$ given $(\vec{\mu}, \vec{\sigma}^2)$ and partitions discussed in Section A.1.1 to distribute our approximate E-step across multiple processors. Given each partition of $\vec{\theta}$, 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 $\log p(\vec{\theta}|\vec{y}, \vec{\mu}, \vec{\sigma}^2)$ using different subsets of $\vec{\theta}$ (corresponding to different blocks) in each iteration. This converges to the maximum of $g(\vec{\theta})$ given $\vec{\mu}^{(r-1)}$ and $\vec{\sigma}^{2\,(r-1)}$.

The approximate E-step considers each block in each of the four configurations, during one iteration. More details are give in Section A.1.2. Within each block $m_1 : m_2$, we maximize $\log p(\vec{\theta}_{m_1:m_2}|\vec{y}, \vec{\mu}, \vec{\sigma}^2, \vec{\theta}_{-(m_1:m_2)})$ numerically via L-BFGS-B or a truncated Newton algorithm (Zhu et al., 1997; **?**); 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., **?**). 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 $\log p(\vec{\theta}|\vec{y}, \vec{\mu}, \vec{\sigma}^2)$ 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 4.

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

Algorithm 4: Approximate EM Algorithm

ALGORITHMIC DETAILS

The algorithm outlined in Section A.1.2 can be implemented on distributed systems with MPI, using the same techniques as the MCMC algorithm presented in Section A.1.1. 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 5.

We this algorithm process a chromosome with 1.5*e*6 base pairs in only 11 minutes using 256 threads on Amazon EC2; a chromosome with 2.4*e*5 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.

Parallel Implementation of Approximate E-Step

```
/* Send conditioning information                                  */
Broadcast $\vec{\mu}^{(t-1)}$, and $\vec{\sigma}^{2\,(t-1)}$ 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 $\vec{\theta}[start : end]$ 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 $\vec{\theta}[start : end]$ from arbitrary worker with tag $b_1$ ;
        Incorporate result into working copy of $\vec{\theta}^{(t)}$ ;
        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 $\vec{\theta}[start : end]$ to last completed worker process with work tag
            attached;
            nStarted++;
    fintq
/* Compute approximate variance, if needed                        */
Compute approximate Var $\left[\theta_k | \vec{y}, \mu_{s_k}, \sigma^2_{s_k}\right]$ using sparse Cholesky
decomposition or diagonal approximation to Hessian;
```

Algorithm 5: Approximate E-Step

A.2 ADDITIONAL FIGURES

A.2.1 REPRODUCIBILITY ANALYSIS—COMPARABILITY OF CLUSTER-LEVEL ESTIMATORS

A.2.2 POWER ANALYSIS—CLUSTER LOCATIONS

Figure A.1 ***(following page)*****:** Joint distributions of of model-based and read-based estimates of cluster-level properties.

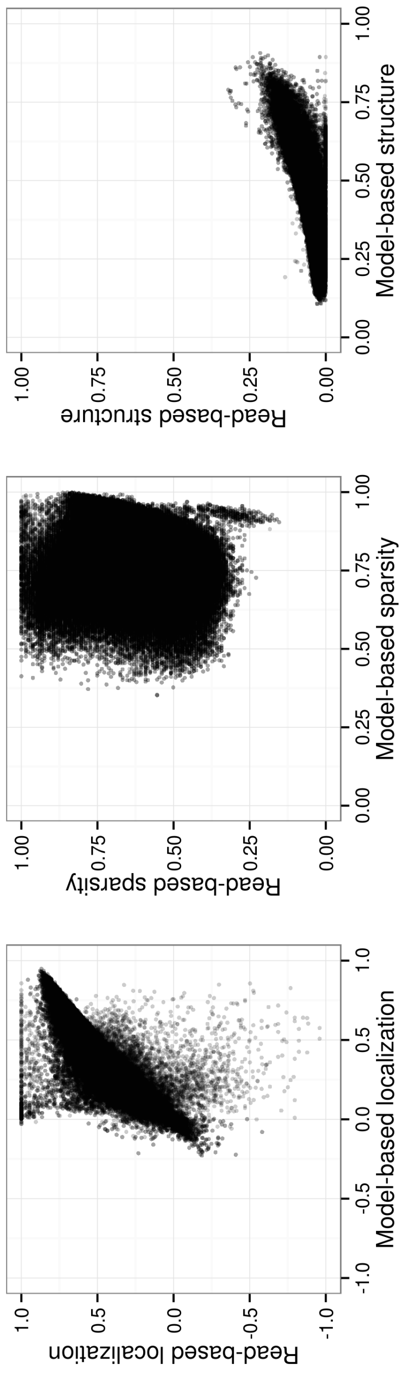


Figure A.1: (continued)

Figure A.2 ***(following page)*****:** 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 A.2: (continued)

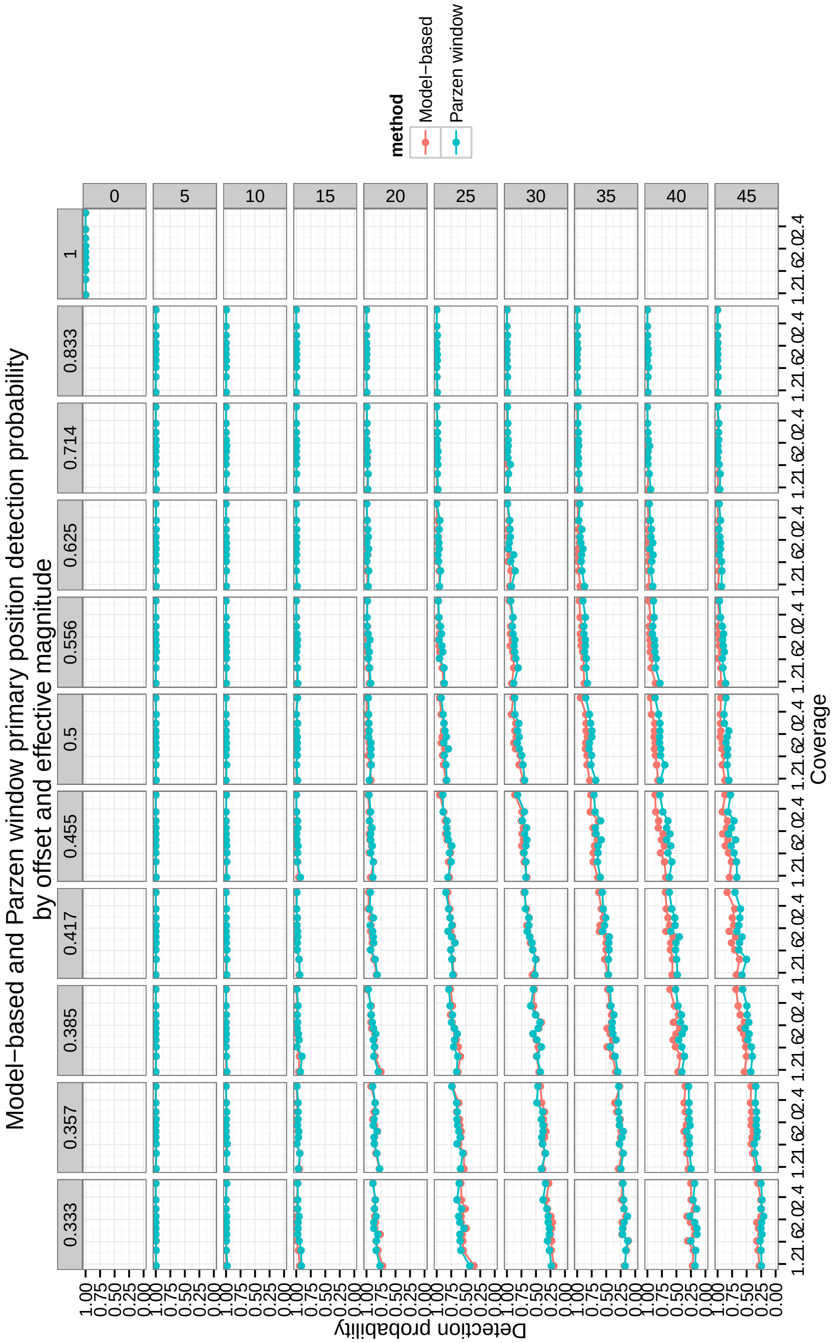
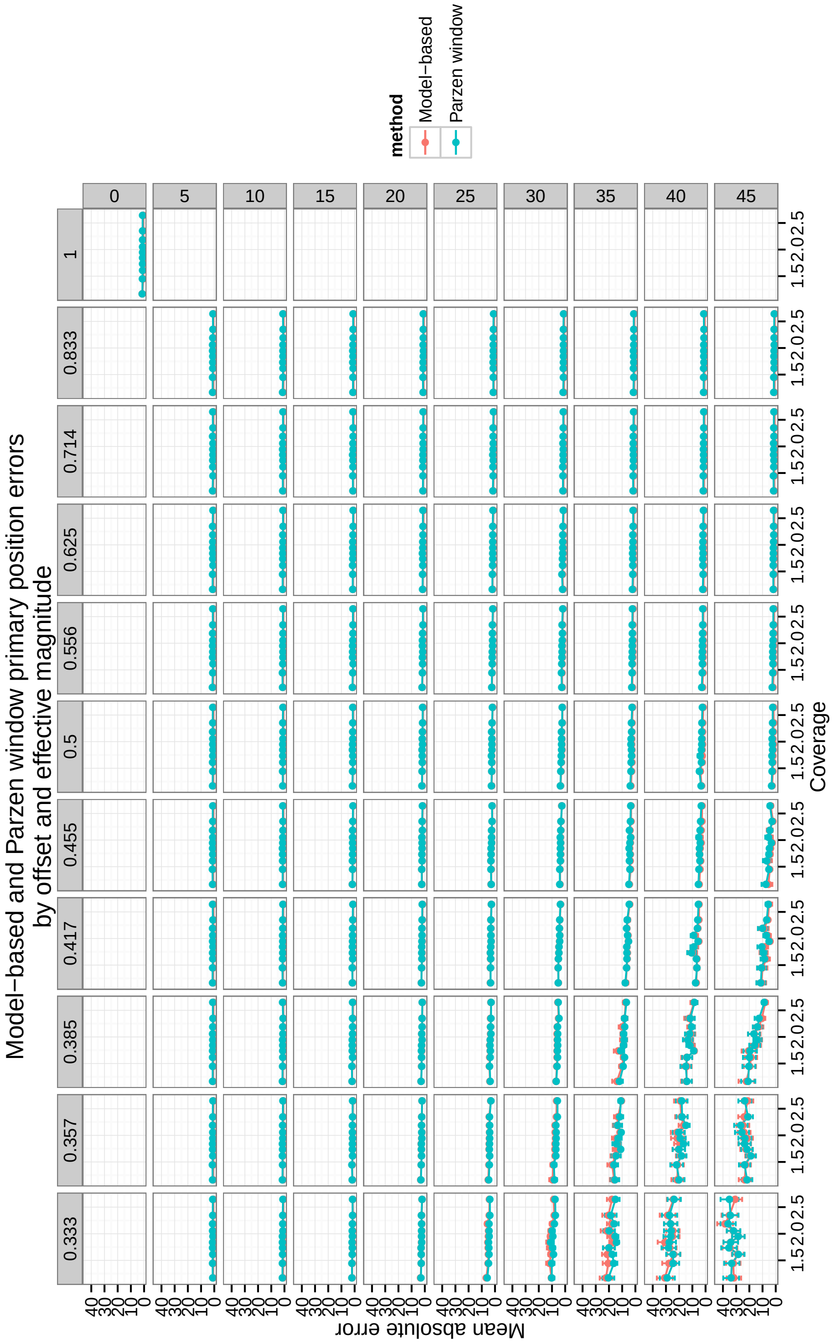


Figure A.3 ***(following page)*****:** Mean absolute position errors for model-based and Parzen window methods vs. coverage by alternative position offset (rows) and effective magnitude of primary position (columns)

Figure A.3: (continued)



A.2.3 Power analysis—local concentrations

Primary positions

Figure A.4 ***(following page)***: 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 A.4: (continued)

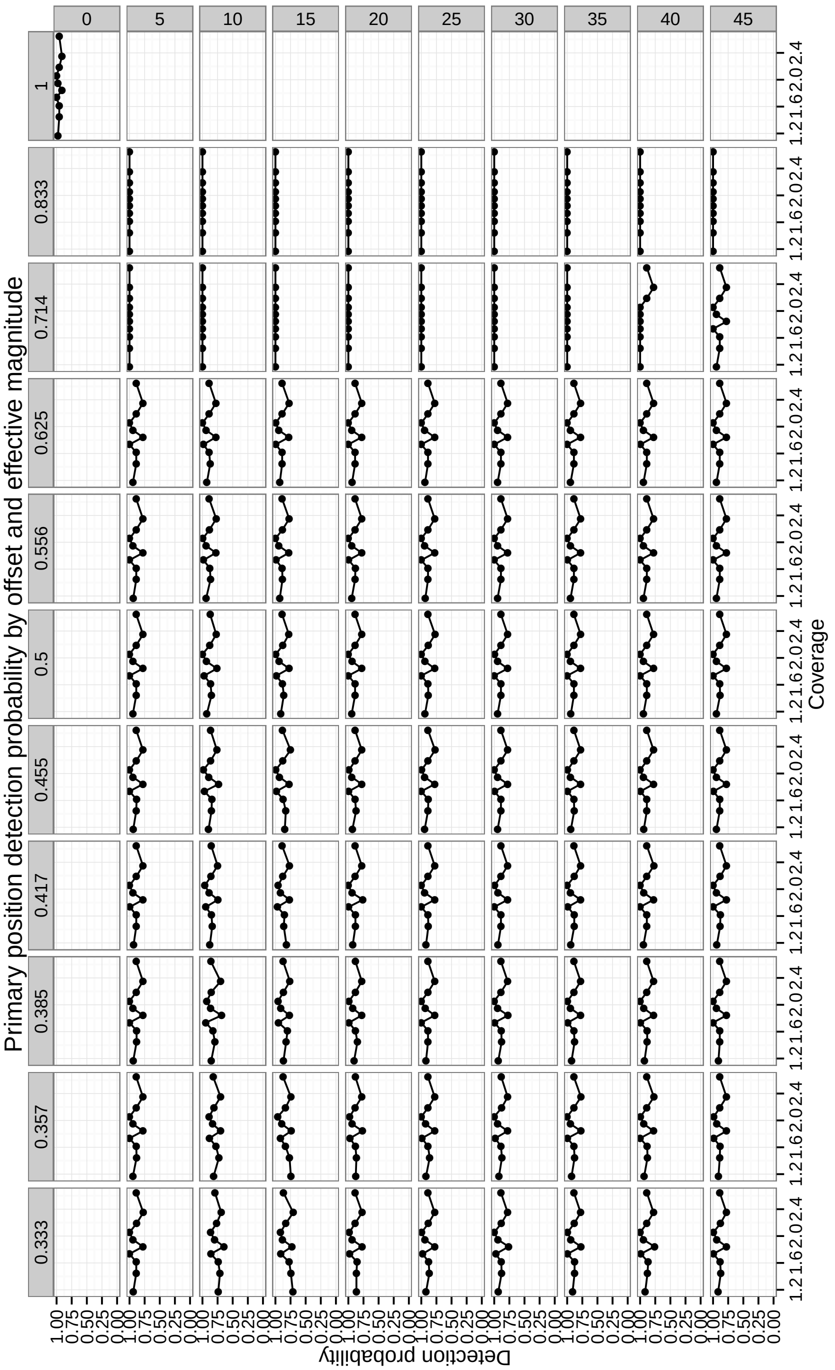
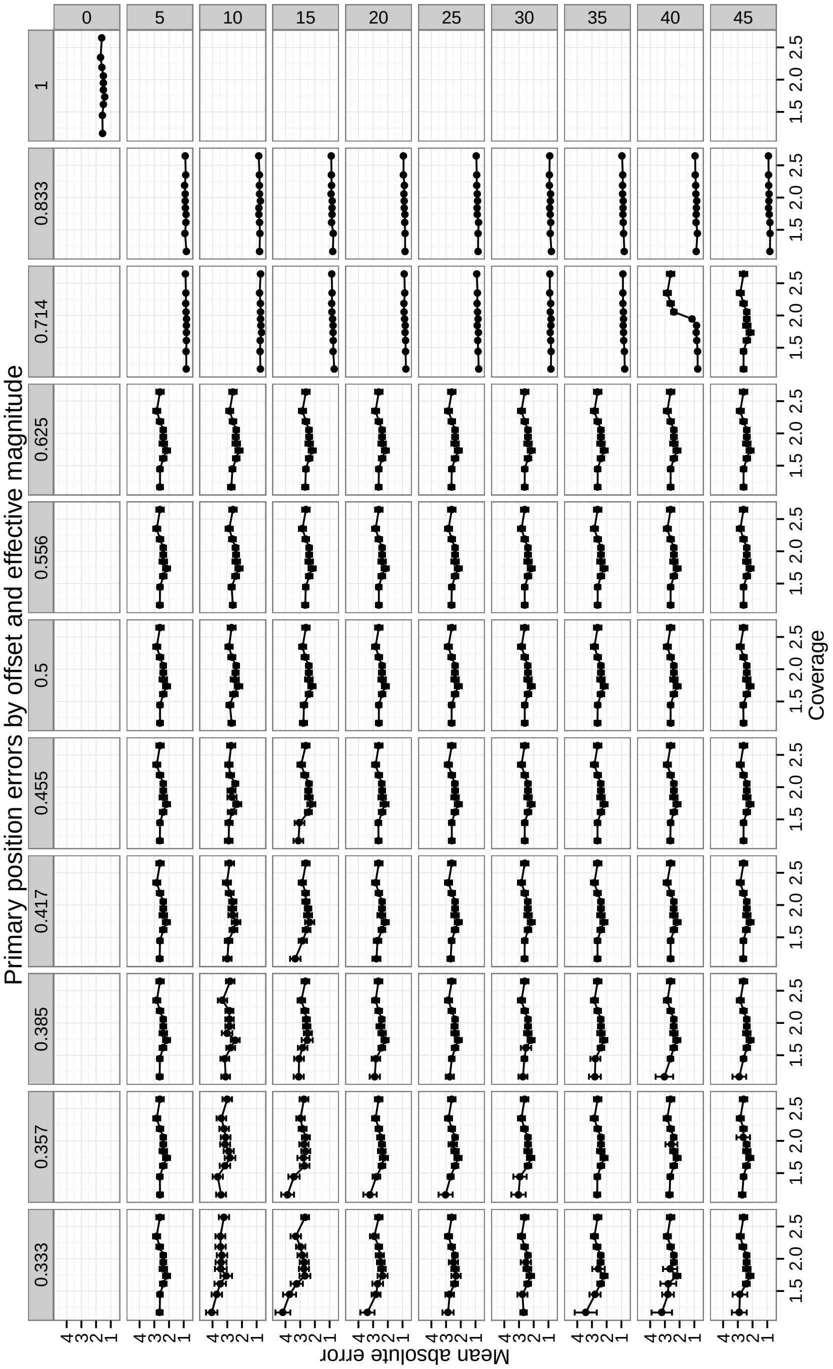


Figure A.5 ***(following page)*****:** Mean absolute position errors of model-based method for individual primary positions vs. coverage by alternative position offset (rows) and effective magnitude of primary position (columns)

Figure A.5: (continued)



Alternative positions

Figure A.6 ***(following page)*****:** 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 A.6: (continued)

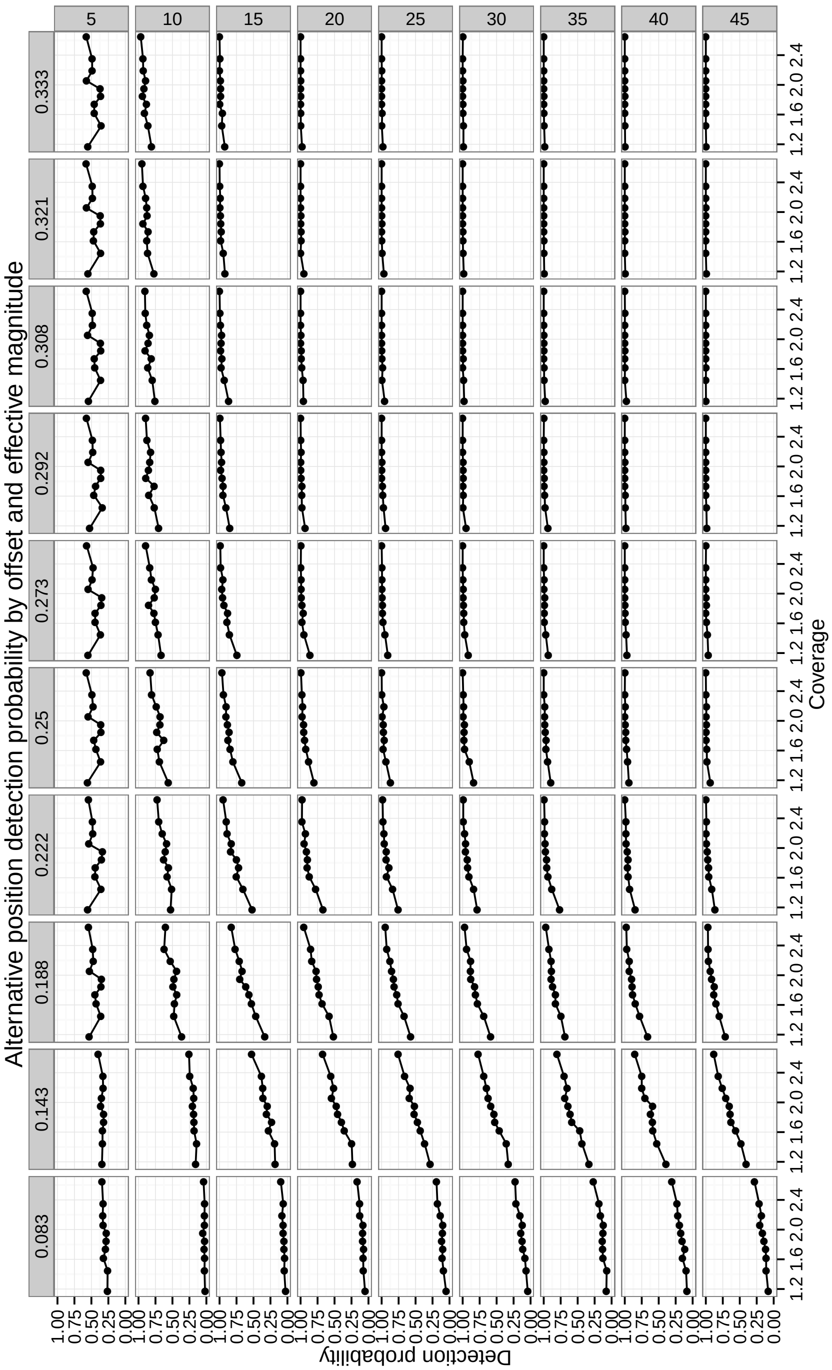
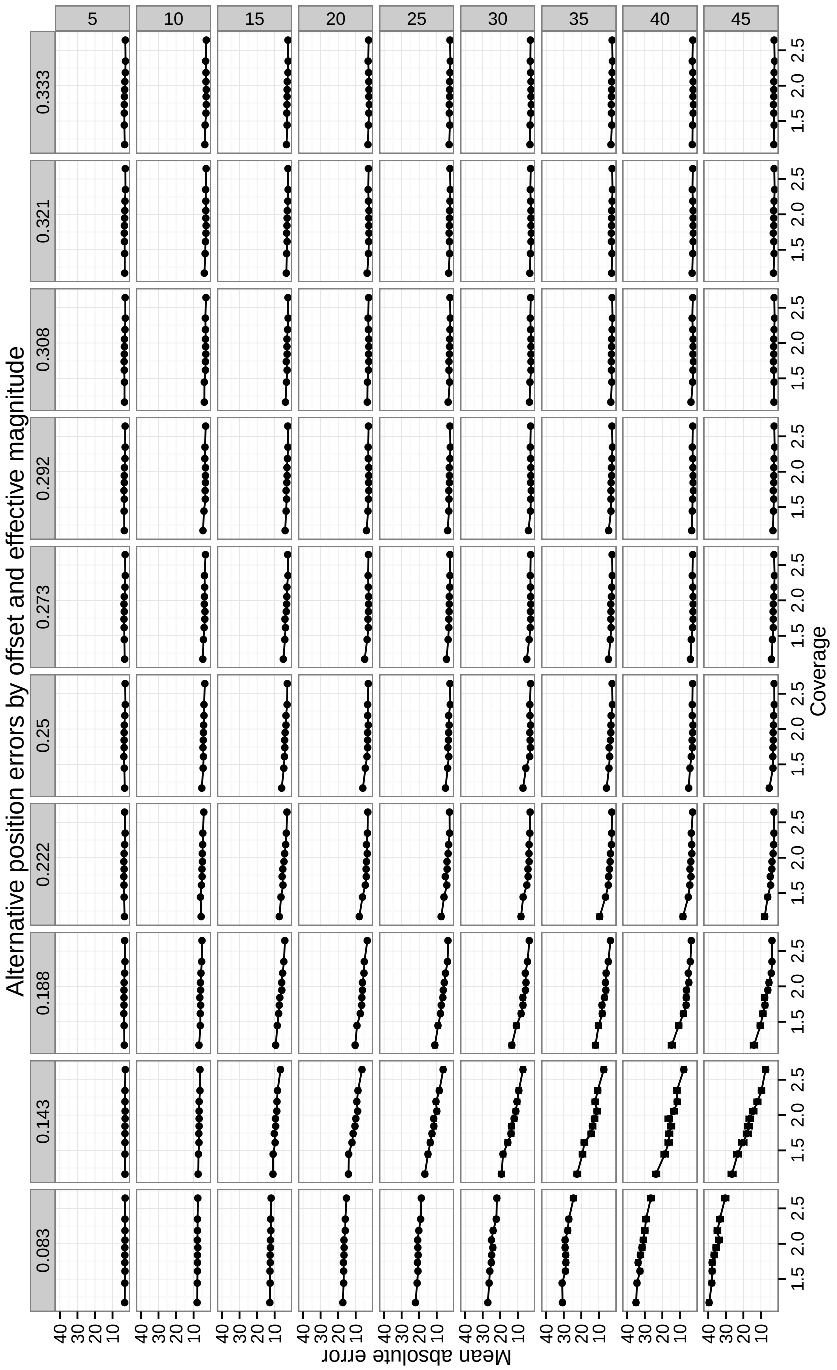


Figure A.7 ***(following page)*****:** Mean absolute position errors of model-based method for individual alternative positions vs. coverage by alternative position offset (rows) and effective magnitude of alternative position (columns)

Figure A.7: (continued)