Title
NORMAL: accurate nucleosome positioning using a modified Gaussian mixture model.

Permalink
https://escholarship.org/uc/item/679347g1

Journal
Bioinformatics (Oxford, England), 28(12)

ISSN
1367-4803

Authors
Polishko, Anton
Ponts, Nadia
Le Roch, Karine G
et al.

Publication Date
2012-06-01

DOI
10.1093/bioinformatics/bts206

Peer reviewed
**ABSTRACT**

Motivation: Nucleosomes are the basic elements of chromatin structure. They control the packaging of DNA and play a critical role in gene regulation by allowing physical access to transcription factors. The advent of second-generation sequencing has enabled landmark genome-wide studies of nucleosome positions for several model organisms. Current methods to determine nucleosome positioning first compute an occupancy coverage profile by mapping nucleosome-enriched sequenced reads to a reference genome; then, nucleosomes are placed according to the peaks of the coverage profile. These methods are quite accurate on placing isolated nucleosomes, but they do not properly handle more complex configurations. Also, they can only provide the positions of nucleosomes and their occupancy level, whereas it is very beneficial to supply molecular biologists additional information about nucleosomes like the probability of placement, the size of DNA fragments enriched for nucleosomes and/or whether nucleosomes are well positioned or ‘fuzzy’ in the sequenced cell sample.

Results: We address these issues by providing a novel method based on a parametric probabilistic model. An expectation-maximization algorithm is used to infer the parameters of the mixture of distributions. We compare the performance of our method on two real datasets against template filtering, which is considered the current state-of-the-art. On synthetic data, we show that our method can resolve more accurately complex configurations of nucleosomes, and it is more robust to user-defined parameters. On real data, we show that our method detects a significantly higher number of nucleosomes.

Availability: Visit http://www.cs.ucr.edu/~polishko

Contact: stelo@cs.ucr.edu or polishka@cs.ucr.edu

© The Author(s) 2012. Published by Oxford University Press. This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
mapped to location the function is equal to the number of sequenced reads that are
et al.
Shivaswamy
(2011). Existing methods in the literature are based on the analysis
eq\sum\text{(e)} represents ‘fuzzy’ nucleosomes
‘blurred’), or how strong is the binding between nucleosomes and
DNA. To address these shortcomings, we propose a method that
determines the accurate position of the nucleosomes independently
from the amount of overlaps in the nucleosomes. This method can
also extract other important statistics about nucleosomes, e.g. the
probability that a nucleosome is actually present, a measure of
nucleosome ‘fuzziness’, and the size of DNA fragments enriched
for nucleosomes.

Here, we propose a parametric probabilistic model for
nucleosome positioning, which we call NOrMAL, for
Nucleosome Mapping ALgorithm. NOrMAL uses Expectation
Maximization (EM) to infer its parameters. To demonstrate the
performance of our method, we report experimental results on
MAINE-seq data for Plasmodium falciparum (Ponts et al. 2010)
and Saccharomyces cerevisiae (Weiner et al. 2014). We compare
the performance of our method against the TEMPLATE FILTERING
(TF) algorithm (Weiner et al. 2014), which is considered the
current state-of-the-art in terms of accuracy and ability to estimate
sizes of the DNA fragments bound to nucleosomes. We also discuss
a fundamental limitation of greedy peak-calling approaches in the
case of overlapping nucleosomes and how our method addresses
this issue.

1.1 Previous work

Several landmark studies have been published in the last few
years on the chromatin structure of model organisms based on the
analysis of genome-wide nucleosome maps. Albert et al. (2013)
Field et al. (2008) MAVrich et al. (2008a,b) Ponts et al. (2010)
Shivawamy et al. (2008) Valouev et al. (2008) Zhang and Puga (2011).
Existing methods in the literature are based on the analysis of
the peaks in the nucleosome occupancy coverages estimated by
mapping nucleosome-enriched reads to the reference genome. The
coverage occupancy profile is an integer-valued function defined
for all genomic locations: given a position \(i\) in a chromosome
the function is equal to the number of sequenced reads that are
mapped to location \(i\). From a probabilistic point of view, the coverage
profile represents a non-parametric distribution of the nucleosome
positions. At the time of writing, the length of the reads obtained by
second-generation sequencing (e.g. Illumina Genome Analyzer) are
limited to about 100 bases and the sequencing occurs in the \(5' \rightarrow 3'\)
direction. In the case of ChIP-seq/MAINE-seq, sequenced reads that
can be uniquely mapped to the positive strand originate from the
left boundary of nucleosome DNA fragments, while reads uniquely
mapped to the negative strand originate from the right boundary
(Fig. 1). Recall that nucleosomes are composed of about 146 bp of
DNA, so if reads are single end and shorter than 146 bp, we expect
observe a peak in the forward and a peak in the reverse coverage
profiles at a distance consistent with the nucleosome size.

The problem of associating a peak in the forward strand with
the correct peak in the negative strand can be difficult in the case
of a large number of complex nucleosome configurations. Some
authors artificially extend the reads in the \(5' \rightarrow 3'\) direction or they
shift the positions of the mapped read position of the forward and
reverse toward the middle of potential nucleosomes. Then, they
combine (e.g. sum) the forward and reverse modified coverages
build a score function. In both cases, they need to determine
the amount of the extension or the size of the shift. In the former
case, the extension should account for the expected length of the
DNA fragments enriched for nucleosomes; in the latter, the shift
should be about half of the DNA fragment size. The problem of this
approach is that no extension or shift that will work equally well
for all nucleosomes in the genome. While one should expect DNA
fragments enriched for nucleosomes to be \(\sim 146 \) bp, reality is that
the digestion process can either leave nucleosome-free DNA in the
sample, or ‘over-digest’ the ends of nucleosome-bound DNA. What
complicates the matter further is that the rate of digestion is sequence
dependent (Allan et al. 2011, Weiner et al. 2011), so nucleosomes in
different genomic locations will end up with different DNA fragment
size. For this reason, it is advantageous to ‘learn’ this information
from the input data. TF (Weiner et al. 2011) is the only method we
know that can handle variable fragment sizes in a specified range,
whereas other methods require users to decide this value in advance.

As said, a variety of peak-calling algorithms have been also
developed (Albert et al. 2008, Field et al. 2008, 2009, Kaplan
et al. 2008, MAVrich et al. 2008a,b, Sasaki et al. 2008, Valouev
et al., 2008). Most of these methods have been proposed for the
analysis of ChIP-chip or ChIP-seq data to determine the position and
strength of the transcription factors binding to DNA. The problem
of detecting transcription factor binding sites is similar to nucleosome
positioning: in both cases we need to infer position of proteins
binding to DNA from the coverage profiles. However, the size of
the nucleosomes is significantly bigger than transcription factor
binding sites, as a consequence the resulting configurations of nucleosomes
can be more complex.

To summarize our experience with existing methods on the
genome-wide nucleosome study of human malaria parasite (Ponts
et al. 2010, 2011), peak calling approaches suffer from a variety
of problems. First, the coverage profile function has to be cleaned
of high-frequency noise, typically via a kernel density estimation
method (Parzen 1962). The type of kernel and the amount of
smoothing can drastically affect the results: too much can merge
adjacent peaks, too little can leave too many noisy artifacts that can
be interpreted as individual peaks. Second, peak finding algorithms
have parameters (like the extension and the shift discussed above)
that are difficult to optimize: a set of parameter can work for a
region of a chromosome but not for another. Third, peak calling
do not properly resolve overlapping nucleosomes. For instance, TF
(Weiner et al. 2011) uses a greedy strategy: nucleosomes are placed

Fig. 1. Nucleosomes are represented by ovals, mapped reads by arrows
(which correspond to \(5' \rightarrow 3'\) prefixes of nucleosome-bound DNA). Coverage
profiles are represented as time series for forward (top) and reverse (bottom)
strands: the line style (solid, dotted and dashed) indicates peaks originating
from distinct nucleosomes; (a) illustrates overlapping nucleosomes; and (e)
represents ‘fuzzy’ nucleosomes
according to the 'best' matching peaks in the score function. Once these strong-positioned nucleosome are assigned, TF ignores any nucleosome that overlaps with previous ones. It is relatively easy to show that for overlapping nucleosomes the greedy strategy does not always return the best overall placement (see Section 3 for details).

2 METHODS

Next, we propose a parametric probabilistic model to find the most likely set of nucleosomes that best 'explain' the mapped reads. We cast this problem in a modified Gaussian mixture model framework. The problem of positioning nucleosomes is then reduced to the problem of learning the parameters of the model and finding the distribution of mixture components, which is achieved via EM.

2.1 A probabilistic model for nucleosomes

We employ a probabilistic model for nucleosome positioning that is described by a set of hidden and observed variables. We use $N$ to denote the number of DNA fragments obtained after MNase digestion. For any DNA fragment $i \in \{1, N\}$, let $x_i$ be the starting position of the 5' end of fragment $i$ (obtained by mapping a corresponding sequenced read), and let variable $d_i \in \{+1, -1\}$ be the strand on which fragment $i$ was mapped (+1 for the positive strand, and −1 for the negative strand). Also, let $\ell_i$ be the length of fragment $i$. If we use variable $m_i$ to denote the position of the center of the fragment $i$, then we have $m_i = x_i + (d_i \ell_i)/2$.

We denote with $X, D, Z$ and $M$ the random variables associated with variables $x_i, d_i, \ell_i$ and $m_i$, respectively. Since the sequencing process is $S \rightarrow \hat{S}$, the value of $X$ is observable by means of mapping a read originating from fragment $i$. Similarly, the strand variable $D_i$ is also observable. Variables $Z_i$ and $M_i$ can be observed directly if sequencing produces paired-end reads, otherwise these variables are hidden. In order to consider the most general case, we only deal with the latter case (single-end reads).

We assume for the time being that the number $K$ of nucleosomes is given. We will discuss how to choose $K$ in Section 2.3. For each DNA fragment $i$, we use a hidden variable $C_i \in \{1, K\}$ representing the nucleosome to which it belongs. Each nucleosome $j \in \{1, K\}$ is described by a set of six variables $(\mu_j, \sigma_j, \Delta_j, \delta_j^+ \Delta_j^-, \pi_j)$, where $\mu_j$ denotes the center position of the nucleosome $j$, $\sigma_j$ is the fuzziness associated with the position of nucleosome $j$, $\Delta_j$ describes the length of DNA fragments associated with nucleosome $j$, $\delta_j^+$ and $\delta_j^-$ represents the variation on fragment sizes for positive and negative strands, respectively, and $\pi_j$ is the probability of nucleosome $j$.

The degree of fuzziness captures the variation of the position of a particular nucleosome in the population of sampled cells. Well-positioned nucleosomes have very low degree of fuzziness. We introduce two variables $\delta_j^+$ and $\delta_j^-$ to model the variation of the fragment size because MNase does not only digest nucleosome-free DNA. Given enough time, it can also digest into the ends of the fragments bounds to nucleosomes, the rate of digestion being sequence dependent (see, e.g. [Allan et al. 2001; Berenbrink et al. 2003]). Since the sequence composition of the 5' end of a DNA fragment can be quite different from the 3' end, we need to have two different variables. The value of $C_i$ is drawn from $(1, 2, \ldots, K)$ with corresponding probabilities $(\pi_1, \pi_2, \ldots, \pi_K)$. Parameter $\pi_j$ models the contribution of $j$th nucleosome to the occupancy level, i.e. what portion of the mapped reads belong to nucleosome $j$.

Our nucleosome model assumes that our random variables are distributed according to a normal distribution. For convenience of notation, we set $\Theta_j = (\mu_j, \sigma_j, \Delta_j, \delta_j^+, \delta_j^-, \pi_j)$ for all $j \in \{1, K\}$, and $\Theta = (\Theta_1, \ldots, \Theta_K)$. First, we assume that variable $M_i$ associated with the center of the fragment $i$ for a particular nucleosome $j$ is distributed as follows

$$P(M_i | C_i = j, \Theta) = \mathcal{N}(\mu_j, \sigma_j^2)$$

where $\mu_j$ represents the center of the nucleosome $j$ and $\sigma_j$ is its fuzziness. Second, we assume that the length $Z_i$ of fragment $i$ for a particular

![Fig. 2. The proposed graphical mixture model: shaded nodes correspond to observed variables, white nodes correspond to hidden variables.](image-url)
Algorithm 1: NONoMAL algorithm

1: (Parameter initialization)
2: \( \mu_0 \leftarrow (\mu_1, \mu_2, \ldots, \mu_K) \), where \( \mu_i \) is uniformly distributed
3: \( \tau_0 \leftarrow (\frac{1}{K}, \frac{1}{K}, \ldots, \frac{1}{K}) \in \mathbb{R}^K \)
4: \( \Theta^{(0)} \leftarrow (\mu_0, \tau_0, \Delta_0, \delta_0, \delta_1, \tau_0) \)
5: (Soft Learning)
6: \( \Theta \leftarrow \text{Learn}(\Theta^{(0)}) \)
7: (Hard learning)
8: for all \( i \in [1, N] \) do
9: \( C_i \leftarrow \text{argmax}_j(P(X_i | \Theta^{(i)})) \)
10: end for
11: Reexecute cluster parameters \( \Theta \)
12: return \( \Theta \)

Algorithm 2: Learn(\Theta)

1: repeat
2: while not converged do
3: \( Q(\theta|\Theta^{(i-1)}) \leftarrow \text{E}_{C_{i,X_0}} \lambda^{(i)}(X|\Theta) \)
4: \( \Theta^{(i)} \leftarrow \text{argmax}_Q(\Theta|\Theta^{(i-1)}) \)
5: \( i \leftarrow i + 1 \)
6: end while
7: for all \( j \in [1, K-1] \) do
8: if \( |\mu_j - \mu_{j+1}| \leq \text{threshold} \) then
9: \( \text{Merge clusters } i \text{ and } i+1 \)
10: end if
11: end for
12: until no clusters were merged
13: return \( \Theta^{(i)} \)

Fig. 3. A sketch of the proposed NONoMAL algorithm

The presence of parameters \( \tau_j \) that correspond to hidden variables \( C_i \) prevents us from solving Equation (6) directly. We estimate \( \Theta \) via EM. In our case, the E step requires computing the posterior probabilities \( P(C_i = j | X = x_i, \Theta^{(0)}) \) of data points \( x_i \in [1, N] \) with respect to the distribution of \( C_i \) given the current estimate of parameters \( \Theta^{(0)} \):

\[
Q(\theta|\Theta^{(0)}) = \text{E}_{C_{i,X_0}} \lambda^{(0)}(X|\Theta) \tag{7}
\]

During the E step, we supplement the missing data in Equation (8) with the expected values under the current parameter estimates \( \Theta^{(0)} \). In the M step, we find new parameter estimation \( \Theta^{(0)} \) by maximizing Equation (8).

\[
\Theta^{(0)} = \text{argmax}_Q(\lambda^{(0)}(\Theta^{(0)})) \tag{8}
\]

It is relatively straightforward to bound the variation parameters \( \mu \) during the EM process to converge to a solution with ‘reasonable’ parameters. We can also easily introduce prior distribution for some of the parameters. For instance, we can specify an expected distribution for DNA fragment sizes \( \Delta_0 \), which can be estimated via gel electrophoresis prior to sequencing.

2.3 Choosing the number of nucleosomes

The method described above assumes that the number of clusters \( K \) is known. The problem of selecting the best value for \( K \) is as challenging as selecting the optimal number of clusters in k-means clustering. One can estimate the number of clusters by looking at the support area of the occupancy coverage, but this will be quite inaccurate because ‘fuzzy’ nucleosomes correspond to wider peaks, and the support area is bigger for them.

Here, we propose a simple but effective heuristic to find \( K \). We start by (1) placing the maximum possible number of non-overlapping nucleosomes uniformly distributed on the chromosome, that is \( K = \text{size of the chromosome)/expected size of a nucleosome} \), where the expected size of nucleosomes is underestimated. Then, (2) we run our EM algorithm until convergence (‘soft learning’). We will then (3) check the distance between the clusters, and merge those that have too much overlap (above a user-specified threshold). In case of multiple overlaps for a nucleosome, we merge it with the closest one. We repeat (2) and (3) until no additional clusters are merged. After a few cycles, we will obtain a set of non-overlapping clusters that best explain the given data points. Overlapping nucleosomes are merged into new ones and then the position of new nucleosomes are estimated from the data.

This procedure will give us a good estimate on the number of clusters as well as a rough estimate of the nucleosome positions. To further improve accuracy for other model parameters, we perform one iteration of ‘hard learning’ by assigning each data point \( x_i \) its maximum probable cluster.

Nucleosome clusters will partition the set of input points, which in turns will allow us to compute their parameters more accurately. The pseudo-code of the algorithm is shown on Figure 3. The running time of NONoMAL is dominated by the running time of Learning step (Algorithm 1, line 6). Observe that the probability that a data point belongs to far-away nucleosomes is close to zero, so one can avoid unnecessary computations by computing updates only for clusters in close vicinity of each point.

The running time is dominated by the heuristic used to find the number of nucleosomes \( K \). In order for the algorithm to scale to eukaryotic genomes, additional optimization steps will have to be implemented. For instance, during the early stage of soft learning (i.e. active cluster merging), the algorithm could be applied to small ‘chunks’ of chromosomes. Then, when the number of merges reduces substantially, the nucleosome maps for each chunk could be combined and algorithm would continue to the hard-learning step.

2.4 Practical considerations

Our method requires users to specify three parameters, namely the threshold for allowed overlap between adjacent nucleosomes, the prior \( \Delta \) on nucleosome sizes and its weight \( \lambda \).

The threshold for allowed overlap can significantly affect the output: the more overlap is allowed, the more nucleosomes can be placed in the current implementation, this parameter has to be specified by the user. The prior \( \Delta \) on the nucleosome size and its weight \( \lambda \) control the propagation of ‘knowledge’ from data points on forward strand to data points on the reverse strand, and vice versa. Based on our experience, if the prior size \( \Delta \) is within 30 bp of the ‘true’ fragment size, then the algorithm is consistent in its output.

Our implementation has some additional internal parameters that we are not expecting users to change. While inferring the parameters of our mixture models, some clusters will tend to cover most of the data points using large variances: a common trick to avoid this from happening is to introduce hard limits on such parameters. Our implementation has range limits for the nucleosome sizes and variance to force the method to converge to ‘reasonable’ nucleosome sizes/variances in the early stage of the iterative process. These parameters have been chosen loose enough so by the end of the iterative process, the limits for nucleosome size and variance are rarely hit, and the output is not significantly affected. The hard-learning step completely ignores those upper limits.

Additional details on parameter selection will be found in the on-line User Manual.

3 EXPERIMENTAL RESULTS

We carried out extensive benchmarking between our proposed method NONoMAL and TF-Extraction et al. [2014]. We selected TF because it is considered the current state-of-the-art. It is the only method that, in addition to inferring nucleosome positions, can extract nucleosome fragment sizes and binding scores. TF differs from the traditional peak-calling algorithms because it does not look for peaks in the coverage profiles, but it places nucleosomes at the peaks of a correlation score matrix. Due to its greedy strategy,
Table 1. The parameters used to generate the reads in Figure 4 and the corresponding output results from TF and NOmMAL.

| Parameter | True value | TF | NOmMAL |
|-----------|------------|----|--------|
| $\mu_1$   | 210        | 208| 206    |
| $\Delta_1$| 130        | 125| 134    |
| $\mu_2$   | 300        | 301| 300    |
| $\Delta_2$| 150        | 149| 148    |

TF has significant limitations when dealing with fuzzy/overlapping nucleosomes, as explained next. The setup for the comparison is as follows. The input parameters for NOmMAL are the prior size of the nucleosome fragments and the allowed amount of overlap between nucleosomes. For TF, we used default parameters unless specified otherwise. The default allowed range of nucleosome sizes for TF is $[100,200]$, which centered around the expected nucleosomes size of $\sim 146$ bp.

Synthetic data: first, we want to illustrate the challenge for existing nucleosome positioning methods to deal with the placement of overlapping nucleosomes. The problem derives from the difficulty in distinguishing two overlapping nucleosomes from the ‘fuzzy’ case. To define precisely this problem, we need to introduce a threshold parameter: if the percentage of overlap between two nucleosomes exceeds the threshold, then they should be considered ‘fuzzy’, otherwise they should be treated as separate overlapping nucleosomes. It is relatively easy to show that the greedy strategy does not always give the optimal nucleosome placement in case of overlapping nucleosomes. To do so, we have created a small synthetic dataset that contains reads corresponding to two overlapping nucleosomes. Although we could have used our own parametric model to generate the synthetic data, to avoid the possibility of giving an advantage to our method, we generated the input data according to the template function described in Weiner et al. (2010). The parameters for nucleosome positions ($\mu_1$ and $\mu_2$) and nucleosome sizes ($\Delta_1$ and $\Delta_2$) that we used to generate the reads are reported in Table 1. Figure 4B illustrates the coverage profile for mapped reads. Observe that there are two peaks on forward and reverse strands, which indicates the presence of two nucleosomes. The percentage of overlap is roughly 35%. In the first case, we allowed such amount of overlap in both TF and NOmMAL (Fig. 4B for TF and Fig. 4C for NOmMAL). Both methods correctly reported two overlapping nucleosomes. Observe the error bars attached to the boundaries of nucleosomes reported by NOmMAL, which indicate the positional variance (or ‘fuzziness’) of the corresponding boundary (each bar has length $\sqrt{\sigma_i^2 + \sigma_j^2}$). TF does not provide such information.

In the second case, when the parameters are set so nucleosomes are not allowed to overlap $\geq 30\%$, only one nucleosome should be reported. Figure 4D and E illustrates the output of TF and NOmMAL, respectively. Observe that now there is a fundamental difference: TF’s greedy strategy reports the presence of the first nucleosome, but then it completely ignores the data corresponding to the second nucleosome. This is an entirely arbitrary choice and the user will not be even aware of this. In contrast, NOmMAL reports one nucleosome positioned near the centroid of the data points and correctly indicates that the variance of the nucleosome boundaries in this case is very high, indicating that this nucleosome should be considered ‘fuzzy’.

In addition, TF’s positioning results are very sensitive to its main input parameter, namely the allowed range for nucleosome fragment sizes. With the default size range $[100,200]$, TF reports one nucleosome (Fig. 4F). When we extend the range to $[40,200]$, TF detects one small nucleosome by incorrectly matching the two strongest (but closest) peaks (Fig. 4F). If we change the range to $[100,300]$, TF reports one large nucleosome, this time matching the outmost peaks (Fig. 4G). Even if we allow a larger overlap, TF will still produce the nucleosome in Figure 4B (data not shown). Since the allowed range for nucleosome size in TF is a hard boundary, the results are very dependent from the choice of this parameter. NOmMAL is more robust in that regard because the prior distribution for the fragment sizes in NOmMAL is ‘soft’ and it can adapt to the data.

Real data: the challenge for nucleosome position inference is that the true positions of the nucleosomes are unknown. The lack of a ‘ground-truth’ makes it very hard to benchmark the existing computational methods. To compare between methods, we can only use conservative indicators. We argue that a valuable indicator is the number of reported nucleosomes. Nonetheless, it is difficult to
argue about performance in objective terms. That is why the first dataset we considered is from *S. cerevisiae* (Weiner et al., 2010). This was the original dataset for which TF was designed. The results of TF on this dataset are assumed to be accurate.

To compare NOxMAL and TF, we used the following setup. The main range parameter for TF was set to $[80, 200]$, which is slightly wider than the default parameters. We did not want to penalize TF, since NOxMAL does not have any hard limits for the nucleosome sizes. For NOxMAL, the main parameter is the prior expected value for the nucleosome sizes: we used 140 bp to hit the middle of the specified range of TF. The threshold value of allowed overlap for both methods was set at 35%. All other parameters were left to default values. The results of both methods are reported in Table 2.

Observe that NOxMAL is slower, but it returns on average 2.6% more nucleosomes than TF.

If we compare the distribution of reported nucleosome sizes (Fig. 5), both methods provide consistent results. To compare how reported nucleosomes are related to each other, we performed a matching procedure. We built a bipartite graph, where a node corresponds to a reported nucleosome (each part corresponds to one of the two methods). The bipartite graph is fully connected, and the weight on edge $(u, v)$ is the squared distance between nucleosome $u$ and $v$: when the distance between $u$ and $v$ exceeded 50 bp, we set the weight to $\infty$. Then we solved the weighted assignment problem between the two sets using the Hungarian method. The distribution of pairwise distances between matching nucleosomes is shown in Figure 6. Observe that the distribution is a unimodal bell-shaped curve with mean and mode having near zero value. The number of matched (common) nucleosomes is 81.44% of the total: 8.29% and 10.27% are unique to NOxMAL and TF, respectively.

While the dataset for *S. cerevisiae* is considered to have relatively stable set of nucleosomes (Weiner et al., 2010), the dataset for the human malaria parasite *P. falciparum* has very dynamic nucleosomes (Ponts et al., 2010). The experiment assumes that cells are ‘synchronized’ at each time-point, but the synchronization is not perfect due to experimental limitations. As a consequence, we expect a large number to nucleosomes to exhibit a ‘fuzzy’ behavior.

First, we performed nucleosome placement with the same setup as with the yeast dataset. The distribution of fragment sizes is quite different in this case (Fig. 6). NOxMAL reports fragment sizes with a mean value of 105 bp and mode value of about 120 bp, whereas TF reports a distribution with mean and mode of about 84 bp. If we perform the matching of the reported nucleosomes, the distribution of distances is much wider than for yeast (Fig. 7). Now only 50% of all detected nucleosomes are in common between two methods. A total of 32.38 and 17.62% are unique to NOxMAL and TF, respectively. As expected, the disagreement between NOxMAL and TF...
and TF is much higher on this dataset, due to presence of a much higher fraction of overlapping/fuzzy nucleosomes.

Two examples of the disagreement between TF and NoMAL are shown in Figure 8. Forward and reverse coverage profiles with extension to 35 bp are shown on top. Nucleosomes are represented by ovals, where the height of each oval represents the confidence score. NoMAL also reports the variance associated with the left and the right boundary, represented with error bars. In Figure 9, the coverage profile around coordinate 800 shows two heavily overlapping nucleosomes that should be reported as one ‘fuzzy’ nucleosome. However, TF reports the position of the nucleosome using the stronger pair of forward/reverse peaks and completely ignores the other pair of peaks. As a consequence, the coordinate of the reported nucleosome is shifted compared to the centroid of the four peaks. NoMAL instead correctly places one nucleosome at the centroid with a relatively high ‘fuzziness’ score. Nucleosome 3 is also quite fuzzy, and it is better placed by NoMAL.

Some disagreement exists on nucleosome 6 as well. The left boundary of that nucleosome detected by TF correspond to a very weak peak. This is due to the fact that TFs placement is based on the correlation score rather than the intensity of the peak. In fairness, both methods assign nucleosome 6 a very low confidence score. Finally, TF detects additional nucleosomes a and b, whereas NoMAL reports additional nucleosome c. All these nucleosomes have low confidence scores. Our method did not report a and b because it explained the data using fuzzy nucleosome 2. NoMAL should have merged nucleosome c to nucleosome 1, but because the overlap did not exceed the chosen threshold those two nucleosomes were not merged.

Figure 9 (bottom) illustrates a more complex example of the coverage profile: even for trained experts placing nucleosomes here would be very challenging. The output of NoMAL and TF are quite consistent for nucleosomes associated to strong peaks. In the regions with high density of peaks, TF tends to place nucleosomes of small sizes (see also Fig. 8) and pack them as tight as possible according to allowed overlapping threshold.

In order to increase the agreement between TF and NoMAL, we tried to extend the range of nucleosome sizes for TF to [20,250]. The new fragment size distributions are shown on Figure 10. Observe that by comparing Figures 8 and 10, the size distribution for NoMAL are the same (only truncated in Fig. 8), while the distribution for TF has changed completely, again pointing out how this range parameter can drastically change the results. Using the extended range, TF was allowed to place smaller nucleosomes so the mode and the mean of the size distribution shifted to smaller values. According to the authors of TF, such small nucleosomes can be due to problems in the experimental procedure, namely overexposing the sample to the MNase digestion. However, the gel electrophoresis analysis shows that the expected size of sequenced fragments in our samples after digestion was about 130 bp in length (without adapters). We speculate that TF's approach might have a problem when the range of admissible nucleosome sizes is too wide, and the algorithm confuses boundaries of neighboring nucleosomes.

![Figure 8](image1.png)

![Figure 9](image2.png)
We described a parametric probabilistic model for nucleosomes for different time-points (TF, parameter in square brackets is the range of admissible nucleosome sizes, bold numbers indicate the maximum)

| Time (h) | TF [80–200] | TF [20–250] | NOmMAL |
|---------|-------------|-------------|--------|
| 0       | 1 720       | 2 031       | 1 934  |
| 6       | 1 491       | 1 826       | 1 720  |
| 12      | 1 461       | 2 158       | 2 043  |
| 18      | 1 185       | 1 665       | 1 537  |
| 24      | 1 440       | 1 910       | 1 766  |
| 30      | 1 723       | 2 229       | 2 443  |
| 36      | 1 701       | 2 514       | 2 788  |

The number of reported nucleosomes for all time-points is shown in Table 3. Observe that for time-points 0–24 h, the number of nucleosomes reported by NOmMAL is between TF with range [80,200] and [20,250]. However, recall that the extended range [20,250] is likely to be unreliable. For time-point 30 and 36 h, NOmMAL identifies a higher number of nucleosomes. The 30 and 36 h marks correspond to the schizont stage of the P. falciparum life cycle. During this stage, the parasites divide, the chromatin compacts and a large number of nucleosomes are added.

## 4 CONCLUSION

We described a parametric probabilistic model for nucleosomes positioning framed in the context on a modified Gaussian mixture model. Our method directly addresses the challenges imposed by overlapping and fuzzy nucleosomes, their detection and the inference of their characteristics. We demonstrated with a synthetic example that the current state-of-the-art method does not properly handle complex overlapping configurations. We have also shown that NOmMAL is significantly more robust to user-defined parameters. On real data, our method detects a higher number of nucleosomes. Although our method is currently slower than TF, the processing time is still modest compared to other steps in the sequencing pipeline and we believe the efficiency of NOmMAL can be improved.

## ACKNOWLEDGMENTS

We thank Prof. Christian Shelton (UC Riverside) for helpful discussions on the EM algorithm. We would also like to thank the three anonymous reviewers for their constructive suggestions.

**Funding**: National Science Foundation [grant DBI-1062301]; National Institutes of Health [grant R01 AI85077-01A1].

**Conflict of Interest**: none declared.

## REFERENCES

Albert,J. et al. (2007) Translational and rotational settings of H2A.Z nucleosomes across the Saccharomyces cerevisiae genome. Nature, 446, 572–576.

Allan,J. et al. (2012) Micrococcal nucleus does not substantially bias nucleosome mapping. J. Mol. Biol., 417, 152–164.

Field,Y. et al. (2008) Distinct modes of regulation by chromatin encoded through nucleosome positioning signals. PLoS Comput. Biol., 4, e1000216.

Field,Y. et al. (2009) Gene expression divergence in yeast is coupled to evolution of DNA-encoded nucleosome organization. Nat. Genet., 41, 438–445.

Kaplan,N. et al. (2009) The DNA-encoded nucleosome organization of a eukaryotic genome. Nature, 458, 362–366.

Le Roch,K.G. et al. (2003) Discovery of gene function by expression profiling of the malaria parasite life cycle. Science, 301, 1503–8.

Mavrich,T.N. et al. (2008a) A barrier nucleosome model for statistical positioning of nucleosomes throughout the yeast genome. Genome Res., 18, 1073–1083.

Mavrich,T.N. et al. (2008b) Nucleosome organization in the drosophile genome. Nature, 453, 358–362.

Parzen,E. (1962) On estimation of a probability density function and mode. Ann. Math. Stat., 33, 1065–1076.

Potts,N. et al. (2010) Nucleosome landscape and control of transcription in the human malaria parasite. Genome Res., 20, 226–238.

Potts,N. et al. (2011) Nucleosome occupancy at transcription start sites in the human malaria parasite: A hard-wired evolution of virulence? Infect. Genet. Evol., 11, 716–724.

Saeki,S. et al. (2009) Chromatin-associated periodicity in genetic variation downstream of transcriptional start sites. Science, 323, 401–404.

Shivaswamy,S. et al. (2008) Dynamic remodeling of individual nucleosomes across a eukaryotic genome in response to transcriptional perturbation. PLoS Biol., 6, e55.

Valouev,A. et al. (2008) A high-resolution, nucleosome position map of C. elegans reveals a lack of universal sequence-dictated positioning. Genome Res., 18, 1051–1063.

Weiner,A. (2010) High-resolution nucleosome mapping reveals transcription-dependent promoter packaging. Genome Res., 20, 98–100.

Zaret,K. (2005) Micrococcal nuclease analysis of chromatin structure. Current Protocols in Molecular Biology, 69, 21.1.1–21.1.17.

Zhang,Z. and Pugh,B.F. (2011) High-Resolution Genome-wide Mapping of the Primary Structure of Chromatin. Cell, 144, 175–186.