Nucleosome Positioning from Tiling Microarray Data

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science

by

Moran Yassour

Supervised by

Prof. Nir Friedman

December 2007

The School of Computer Science and Engineering
The Hebrew University of Jerusalem, Israel

Abstract

The packaging of DNA around nucleosomes in eukaryotic cells plays a crucial role in transcriptional regulation, *e.g.*, by altering the accessibility of short transcriptional regulatory elements. To better understand transcription regulation, it is therefore important to identify the position of nucleosomes in 5-10bp resolution. Toward this end, several recent works measured nucleosomal positions in a high-throughput manner using dense tiling arrays.

Here we present a fully automated algorithm to analyze such data. Using a probabilistic graphical model, we suggest to improve the resolution of the nucleosome calls beyond that of the microarray platform used. We show how such a model can be compiled into a simple HMM, allowing for a fast inference of the nucleosome positions, without any loss of accuracy.

We applied our model to nucleosomal data from mid-log yeast cells reported by Yuan et al. [2005] , and compared our predictions to those of the original paper, to a more recent method that uses five times denser tiling arrays [Lee et al., 2007] , and to a curation of literature-based positions. Our results suggest that by applying our algorithm to the same data of Yuan et al., we were able to trace 13% more nucleosomes, and increase the overall accuracy in about 20%. We believe that such an improvement opens the way for a better understanding of the regulatory mechanisms controlling gene expression, and how they are encoded in the DNA.

Acknowledgments

First I would like to thank my advisor Nir Friedman, who taught me so much in the past two years, about how to address my questions, and more importantly, which questions should I ask. For exposing me to the academic world, and enhancing my critical point of view on every new work I see. I was fortunate to have the opportunity to work with an advisor that always has suggestions in both the biological and computational perspectives. I am thankful to Tommy Kaplan, who was in many ways my second advisor, and proved to me time and time again that two thinking minds are always better than one. I can't imagine our lab without you. I would also want to thank all the members of Nir's lab, especially Ariel Jaimovich who always finds the time to help me , Naomi Habib who is always there for me when I need it and Ofer Meshi who is very patient with all the questions I lay upon him. Finally, I would like to thank my parents and sisters who keep pushing me in the right direction, and help me along the way. Last but not least, I thank Liron who is beside me all this time, and always believes in me, even when I don't.

Contents

Chapter 1

Introduction

The DNA in our cells contains all of our hereditary information, and is literally the blueprints of our body. According to the Central Dogma of Biology, DNA is transcribed into the RNA, which is translated in turn to proteins. These proteins eventually carry inter-cellular signals and perform most tasks in the cell.

All the cells in an organism share the same DNA, but we can still observe differences, for example in gene expression, between cells in different tissues and under different conditions. Each cell controls the transcription of the genes by applying a complicated regulation plan. Once a gene is transcribed, additional regulations may occur, and then it will be translated into a protein. The cell's transcriptional regulation plan dictates which genes will start this process, and thus has an important role in determining the cell protein content and behavior.

The basic structure of a gene, as shown in Figure 1.1, demonstrates the different functional segments of a gene. In the transcription proccess, the RNA polymerase binds to the DNA at the promoter, and procceeds along the gene, while transcribing it. The regulatory segment of the gene is called promoter, and is usually located upstream to the transcription start site. Additional proteins called transcription factors, bind to the promoter and can either facilitate or repress the binding of the polymerase to the promoter. Transcription factors execute the regulation plan of the cell in both a positive and a negative manner. Many studies have shown the connection between the binding of the transcription factor to the activity of the gene (*e.g.*, Latchman [2005]). Other factors take part in this regulation, like the packing of the DNA.

In eukaryotic cells the DNA is packed within the nucleus where it is wrapped around protein complexes called nucleosomes, such that each nucleosome is surrounded by roughly 147 DNA bases [Luger et al., 1997]. This

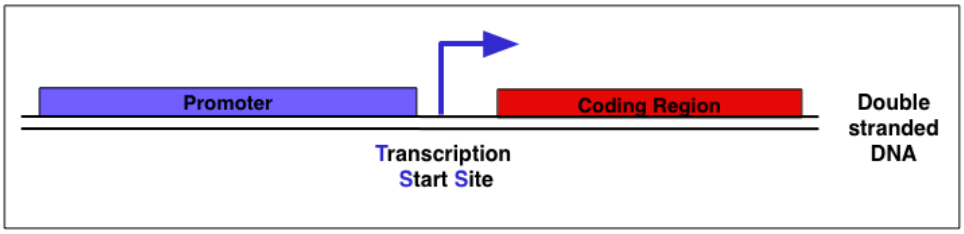


Figure 1.1: Illustration of a gene's spatial structure. On the left is the regulatory sequence, called promoter, on the right is the coding region which will eventually be translated into a protein, and in between is the transcription start site (TSS), where the RNA polymerase will start transcribing this gene.

packaging facilitates the storage and organization of the long eukaryotic chromosomes. The nucleosome is a protein complex of four types of histones, which are among the most conserved proteins known today. The DNA is tightly wrapped around this complex, as shown in Figure 1.2, and this packaging affect the accessibility of the DNA to other factors. The packaging plays a crucial role in regulation of DNA-related processes by modulating the accessibility of DNA to regulatory proteins. Specifically, *linker DNA* regions between nucleosomes are exposed to binding of transcription factors that can thereby affect the expression of nearby genes [Venter et al., 1994, Lee et al., 2004]. It has also been reported by Lee et al. [2004], Yuan et al. [2005] that upstream to the transcription start site there is a region which is usually unoccupied by a nucleosome. This region might promote the binding of transcription factors and RNA polymerase and is called Nucleosome Free Region (NFR). As the regulatory binding sites are typically short (6-20bp), knowledge on the exact location of nucleosomes along the DNA is crucial for understanding the transcriptional blueprints embedded in the DNA.

To find nucleosome locations in a small-scale manner, people have used DNA footprinting to find the location of each nucleosome in their region of interest [Venter et al., 1994], or Chromatin Immunoprecipitation for a certain histone modification with selected PCR primers [Reinke et al., 2001]. We are interested in finding the nucleosome locations in a genome-scale manner, and for that reason we need to use a more elaborate genomic protocol.

To estimate the exact position of nucleosomes along the DNA in yeast cells, we analyzed the tiling microarray data of Yuan et al. [2005]. In this work, MNase assay was used to digest linker DNA regions resulting in mononucleosomal DNA fragments of length $\sim$150bp. These fragments

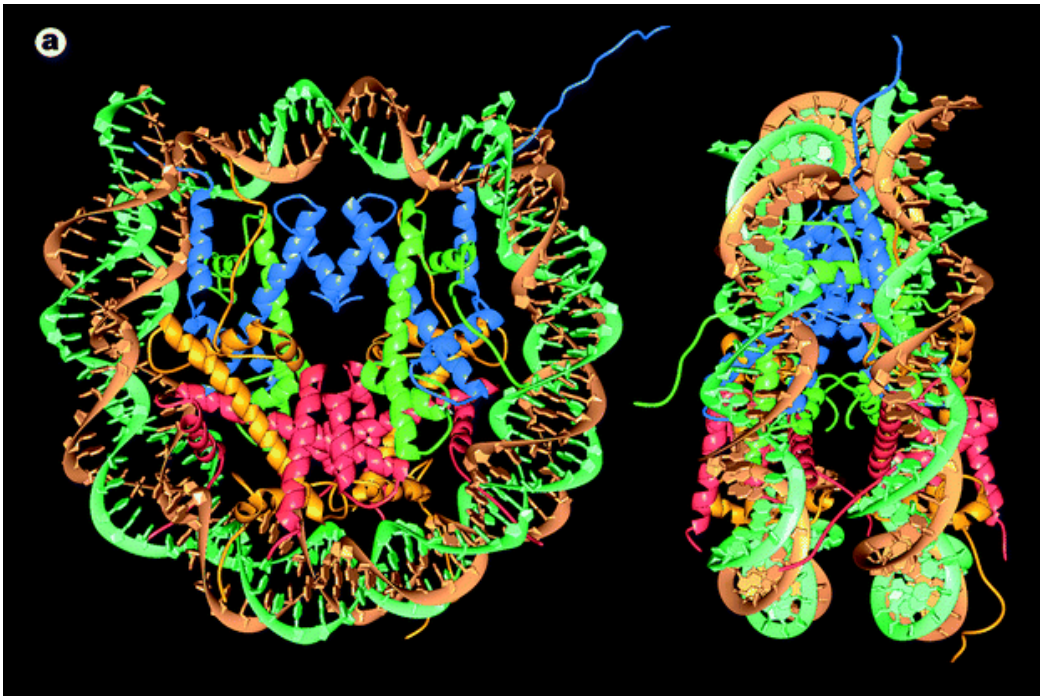


Figure 1.2: The three dimensional structure of the nucleosome with the DNA wrapped around it, as reported by Luger et al. [1997]. Shown are views from above the helix (right), and from the side (left). Also shown are all the histones in the nucleosome complex: H2A (yellow), H2B (red), H3 (blue) and H4 (green). The double helix DNA is shown in green and orange, around the nucleosome.

were then labeled with fluorescent dye and hybridized to microarrays against a genomic DNA reference, see Figure 1.3 for more details. Yuan *et al.*'s microarrays were designed with overlapping 50bp probes tiled every 20bp across the entire *S. cerevisiae* chromosome 3 and additional regions of interest, covering about 4% of the yeast genome [Yuan et al., 2005] .

In this work, we present a fully automated computational method to identify nucleosome positions based on the raw output of microarray measurements of MNase-based assay (*e.g.*, Yuan et al. [2005]). Our emphasis is on improving the resolution of these nucleosome calls beyond that of the microarray platform used. We do so using a probabilistic graphical model that describes how probe values depend on the exact nucleosome positions. We applied our model to nucleosomal data from mid-log yeast cells reported by Yuan et al. [2005], and compared our predictions of nucleosome calls to the original study, to those of a more recent high-throughput method that

uses a higher resolution tiling array [Lee et al., 2007], and to a curation of literature-based positions [Segal et al., 2006]. Our results suggest that by applying our algorithm to the same data of Yuan *et al.*, we were able to trace more nucleosomes, and increase their accuracy.

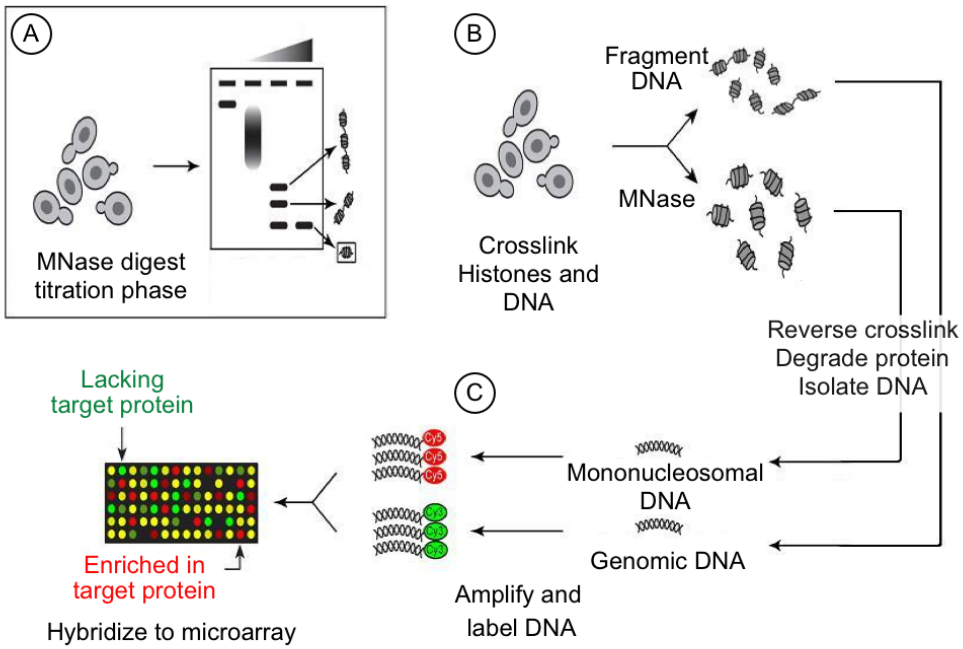


Figure 1.3: MNase chip protocol, as used by Yuan *et al.* , taken from Liu [2006]. **(A)** MNase is used to cut the linker DNA. In the titration phase the correct concentration of MNase is found in order to cut each linker segment. The appropriate concentration will result is a single band in the gel, corresponding to a single nucleosome. **(B)** Formaldehyde is used to fixate all proteins to the DNA, and isolate the DNA with its bound proteins. The DNA is divided into two fractions, one is sonicated to break the DNA at random locations, and the other is treated with MNase such that only mononucleosomal DNA wrapped around nucleosomes is left. DNA is then isolated from both fractions. **(C)** PCR is used to amplify the DNA from both fractions, while labeling the mononucleosomal fraction with the Cy5 dye, and the genomic fraction with the Cy3 dye. Both fractions are now mixed, and hybridized on the array. A red spot means that more mono-nucleosomal DNA than genomic DNA was hybridized on this spot, and the opposite goes for a green spot.

Chapter 2

Models

After performing the MNase chip method, we have genomic data on the presence of nucleosomes, and now we would like to extract the nucleosome positions from these data. Here I present a few methods which address this issue.

To analyze these data, Yuan *et al.* used a hidden Markov model (HMM), that labeled each probe as either a nucleosome or a linker, which defines the emission probability of the hybridization value (shown in Figure 2.1). To perform exact inference on this model, which is to find the most likely assignment to all the hidden variables given the probe values, Yuan et al. [2005] used the Viterbi algorithm [Rabiner and Juang, 1986] and received the *maximum a posteriori* (MAP) assignment.

There are a few problems with the suggested model, the first is that some nucleosomes exhibit a lower hybridization signal than others (probably due to occupancy differences between cells) and the second is the model resolution. Since they decide on each probe whether it is inside or outside a nucleosome, they have misclassifications on the boundaries of the nucleosomes, as the hybridization value fits neither the nucleosome nor the linker's expected values. The HMM nucleosome calls from the Viterbi algorithm were then hand-curated to correct for what they perceived to be missing or wrong nucleosome calls. Yuan *et al.*'s output defines a nucleosome by a set of probes, and thus has inherent resolution of 20bp.

2.1 Simple HMM

To deal with both issues, we developed a fully automated method to analyze the exact same data of Yuan *et al.* with a more detailed model. This allowed us not only to predict the nucleosome locations in higher resolution, but also

to automatically predict the occupancy level of each nucleosome.

2.2 Detailed Model

When we consider the form of the original data (shown in Figure 2.2), we realize that the probes at the boundaries of nucleosomes will have partial hybridization. Indeed, one can observe that the flanking probe values are lower than these in the middle of the nucleosome. Another problem in these data (as in many ChIP arrays) is of trends. Namely, some nucleosomes have higher enrichment values for all their probes while other have lower ones (*e.g.*, the leftmost nucleosome in Figure 2.2 has lower value than the other two nucleosomes). This can be due to differences in nucleosome occupancy, or to differences in the background distribution [Pokholok et al., 2005]. Yuan *et al.* dealt with these "coordinated" changes by re-learning the HMM parameters for local sequence windows. Moreover, the results were then corrected manually.

To address these issues, we define an algorithm which is both fully automatic and has high resolution output (so it can deal better with probes on the boundaries of nucleosomes differently). To double the resolution (from 20bp to 10bp), we present the following graphical model (Figure 2.3). Each 10bp of the genome are represented by a (hidden) variable, indicating whether a nucleosome is present at that locus, and if so, the relative position of the DNA within this nucleosome (S_i variables in Figure 2.3). Specifically, since the length of a nucleosome is about 147bp, and the resolution of the S_i variables is once every 10bp, the possible assignments of S_i are 0 for linker regions, and $1, \ldots, 14$ for nucleosomal regions. This layer of variables has the structure of a sparse HMM, according to the state diagram shown in Figure 2.4.

Thus, the only parameter here is θ which governs the expected length of a linker. To allow for shorter nucleosomes (less than 140bp), as often appear in the raw data, we added the exceptions allowing transition from state 7 to states 8, 9 and 10, which add two more parameters to the model (a and b respectively). These parameters account for observing nucleosomes of length 140bp, 130bp and 120bp. We learned all these parameters using the hand-curated nucleosome calls of Yuan et al. [2005], with regards only to the well-localized nucleosomes (without taking the fuzzy ones into consideration).

To connect this layer to the 50bp probes, we introduce an additional layer of variables, which calculates for each probe the percentage covered by a nucleosome (C_i variables, whose possible assignments are 0% coverage, 20%, ..., 100% coverage). The coverage is a deterministic function of the corresponding S_j values. The assignment of the C_i variable, which is the

percentage of this probe covered by a nucleosome, will eventually affect the expected hybridization value of this probe, as 100% coverage will produce a higher value than 80% and so on.

To account for nucleosomes in different occupancies and to handle global trends in the raw data baseline, we included an additional layer of variables we call the *occupancy variables* L_i, which together with the *max-coverage* variables (C_i) determine the likelihood of the measured probe values P_i. We require that the values of the L_i variables remain fixed within nucleosomes. Thus, these variables create a dependency between the measured values of probes in the same nucleosome. To capture this intuition, each L_i depends on the previous L_{i-1} and the state variables in the intermediate region. If these are within nucleosome, then $L_i = L_{i-1}$, otherwise, L_i is chosen from a prior over occupancy levels.

The emission probability of the model $P(P_i \mid C_i, L_i)$ is of a normal distribution, whose parameters are defined by the the parent variables (C_i and L_i). To learn these parameters, we assigned each of Yuan's nucleosomes an occupancy level, and then learned both the emission and transition probabilities. An example of the normal distributions used is shown in Figure 2.5.

2.2.1 Approximate Inference

The model, as presented in Figure 2.3, is densly connected (especially due to overlapping probes causing loops between the S and the C variables). This makes straightforward exact inference of the S variables (*e.g.*, using a clique tree) extremely time consuming. One way of extracting the positions from this model is to use approximate inference methods.

To obtain an approximated MAP assignment to our variables, we applied the loopy belief propagation algorithm [Murphy et al., 1999] to our model. This is an iterative algorithm that passes messages between the cliques in the model on the potentials of their shared variables, in order to calculate the marginal probabilities on each variable. If an incoming message has changed the beliefs of the clique's variables, then all messages departing from this clique enter the queue, to be re-calculated in future iterations. The algorithm converges if the queue is empty, meaning there are no more messages to send. The loopy belief propagation algorithm is not guaranteed to converge, but if it succeeded to converge, we find its results to be a good approximation. Unfortunately, in our case, the algorithm did not not always converge, even when trying different scheduling methods on the messages in the queue.

The problem was that some genomic locations had a clear signal of nucleosomes positions, while others had a more vague signal and we thought this prevented the converging of the loopy algorithm. When trying to prove this

theory, we counted the number of messages passed in each genomic location. This plot is shown together with the raw data in Figure 2.6. We can clearly see that indeed a certain genomic location at 70,000-71500 does not converge, while all other locations have a very low number of messages, meaning the algorithm has converged there. This means the algorithm encounters a problem at this location, and basically cannot decide on the nucleosome locations. To further validate this, we chose a few messages from this area, and observed their behavior over time. In Figure 2.7 we show the oscillations in the messages content passed in this area. In Figure 2.7(a) the message oscillates between a nucleosome and a linker in this area, and does not succeed in deciding either way. These oscillations diffuse through all the levels of the model, *i.e.*, the message over the coverage variable shown in Figure 2.7(b) oscillates between 0% and 100% in respect to the previous messages on nucleosomes or linker.

After understanding the convergence problem, we allowed the loopy algorithm to run for a certain amount of time, and then according to the number of messages still in the queue, decided for each location whether it had converged or not. In this way, we can control the amount of time the algorithm runs, and then filter the divergent results. Generally, this approach can be applied to many other convergence problems, as long as divergence in one area does not affect the convergence of another area. In our model this is true due to the Markovian property along the axis of the genomic location, once we know the close by nucleosome positions, we are not affected by the nucleosomes located further away.

We were still not pleased with our results, because there were some areas where we could not predict nucleosome locations. Furthermore, in the areas where we predicted nucleosome location, this was only the approximate MAP assignment, and not the exact one. There are also other methods which find the approximated MAP, like Weiss et al. [1999], but we wanted the posterior as well as the MAP, and we also came up with a much better solution.

2.3 Model Compilation

We noticed that due to the sparsity (and in some cases deterministic nature) of the conditional probability distributions, we can drastically improve the performance of exact inference on our model.

We developed an automated algorithm to compile a graphical model such as the one in Figure 2.3 into a simpler HMM (shown in Figure 2.8(b)). The algorithm proceeds in several stages. First, we define the sets of variables that separate previous hidden variables from the observation. Here, this includes

the S_j variables connected to C_i, and L_i connected to the probe P_i, as shown in blue in Figure 2.8(a). We define a new variable X_i whose state is the cross product of these variables. In our current model, the value of X_i is in the space $[0-14] \times [0-14] \times [0-14] \times [0-14] \times [0-14] \times [0-3]$, representing five relative positions in nucleosome states, and one level state for the probe. This state space is enormous (approximately 3 million states), which leaves us with the same problem as before, since exact inference is extremely time consuming. However, when looking closely at the model, we can observe that due to the nature of the detailed model, the number of states can be reduced drastically.

We start by eliminating states that are impossible due to the conditional probability of variables agglomerated within X_i. For example, if S_{10} the parent of S_{20} is in X_1 (see Figure 2.8(b)), then if $P(S_{20} \mid S_{10}) = 0$, the value of X_1 is unattainable. Since our original HMM over the S_j variables is very sparse, this results in a massive reduction in the state space of X_i. The elimination step has tremendously decreased our state space from about 3 million to only 132 states.

Once we have these states, we can define a transition probability between them. This transition is built by computing the conditional probability in the original model. In our case,

$$\begin{aligned} &P(X_2 = \langle s_6, s_7, s_8, s_9, s_{10}, l_2 \rangle \mid X_1 = \langle s_1, s_2, s_3, s_4, s_5, l_1 \rangle) = \\ &\quad 1\{\langle s_3, s_4, s_5 \rangle = \langle s_6, s_7, s_8 \rangle\} \cdot P(S_{50} = s_9 \mid S_{40} = s_8) \cdot P(S_{60} = s_{10} \mid S_{50} = s_9) \\ &\quad \cdot P(L_2 = l_2 \mid L_1 = l_1, S_{30} = s_4, S_{40} = s_5) \end{aligned}$$

Since the state X_i contains all the parents of P_i, the emission probability is exactly as it was in the original model.

In the final step, we perform an additional step of simplifying the model. We say that two states of X_i are *equivalent* if they share the same transition and emission probabilities. Since the transition probability is determined by the last three variables of the state, all states matching $\langle \cdot, \cdot, s_3, s_4, s_5, l_i \rangle$ share the same transition probability. So, any states that obey this rule, and also share the same emission probability are equivalent (*e.g.*, $\langle 8, 9, 10, 11, 12, l_i \rangle$, $\langle 7, 9, 10, 11, 12, l_i \rangle$, $\langle 6, 7, 10, 11, 12, l_i \rangle$). It is easy to prove that merging two equivalent states does not change the likelihood of the observations, as this is an instant of *state abstraction* [Friedman et al., 2000]. We thus repeatedly merge equivalent states, updating the transition probability (which can cause other pairs of states to become equivalent), until all states are non-equivalent to each other.

After finishing this process for the model of Figure 2.3, we end up with an HMM of only 100 states. Since we have been able to reduce the state

space drastically, we can now perform exact inference in a straightforward manner. We then applied the Viterbi algorithm on this model to obtain the MAP assignment, and the forward-backward algorithm to obtain the posterior distributions [Rabiner and Juang, 1986]. Using these results we can now answer queries about the original variables in the model, and locate the exact nucleosome positions.

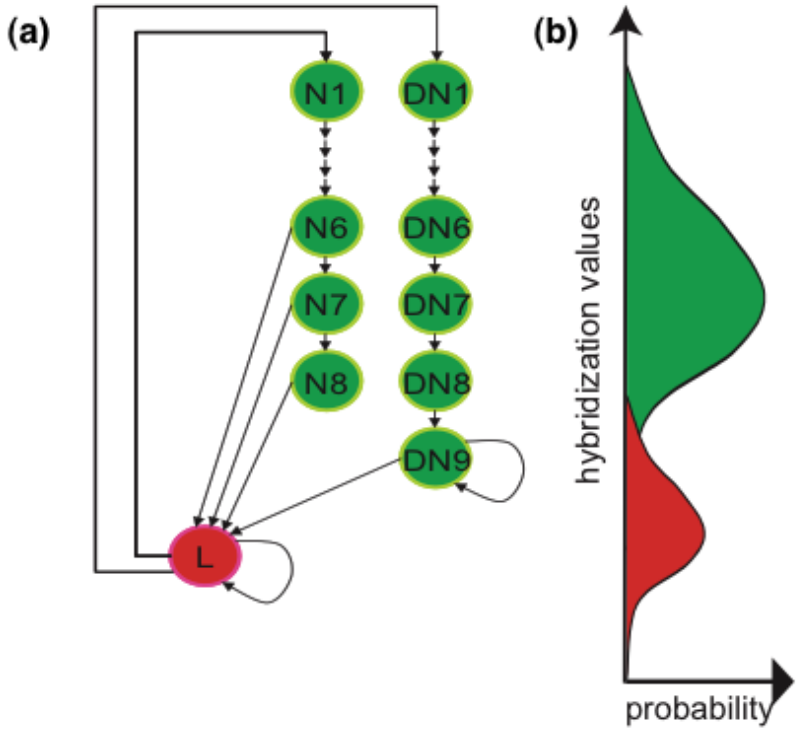


Figure 2.1: Yuan *et al.*'s hidden Markov model, taken from [Yuan et al., 2005] **(a)** The scheme of the model, where green variables represent probes inside a nucleosome, and the red variable represents linker DNA. Well positioned nucleosomes are expected to cover about 6-8 probes (N1-N8) which have a high hybridization ratio, and delocalized nucleosomes might span over 9 or more probes (DN1-DN9), where the return arrow on DN9 allows for longer nucleosomes. In the same manner, the return arrow on the linker variable (L) allows for variable size linkers. **(b)** Model parameters. All probes inside a nucleosome are expected to have higher hybridization ratio, estimated by the green normal distribution, and the linker probes exhibit a lower hybridization ratio, estimated by the red normal distribution.

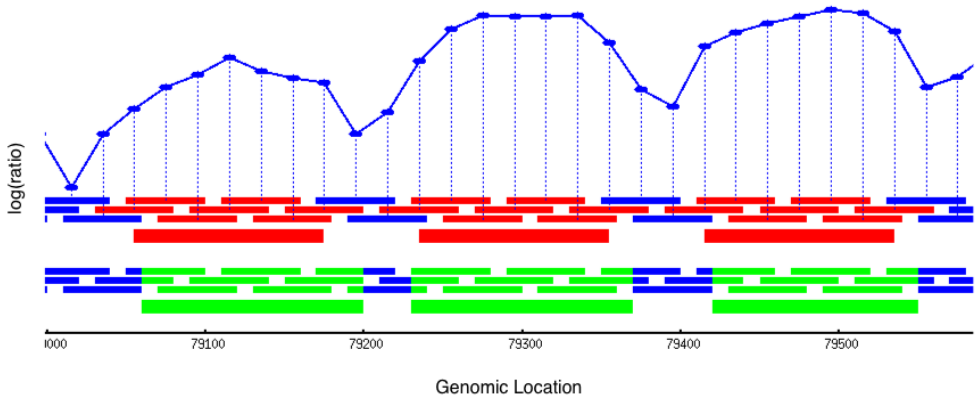


Figure 2.2: Raw data (blue line) from Yuan *et al.* shown on 600bp of chr3 (79,000-79,600), mapped onto probe locations. Top: raw log-ratio of nucleosome occupied DNA against genomic DNA. Bottom: design of tiling array, where each rectangle denotes the coverage of a probe and the vertical line map it to its value. These probe locations were marked with coverage based on Yuan *et al.*'s predictions (red rectangles), where each probe is assigned either to be covered (red) or not (blue). Below, is a representation of our calls for the same region (green), and the description of partial coverage of probes by nucleosomes.

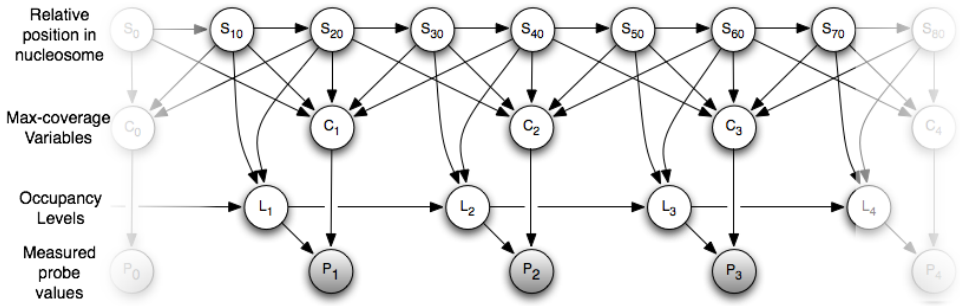


Figure 2.3: Graphical model. The S_i variables report the position of a genomic locus with regard to overlapping nucleosome (in a 10bp resolution), or 0 in case of a linker DNA region. The C_i variables hold the the maximal coverage of a probe by a nucleosome, as reported by the relevant S_j's. L_i's are the inferred occupancy levels for any probe, and P_i's are the probes' measured values.

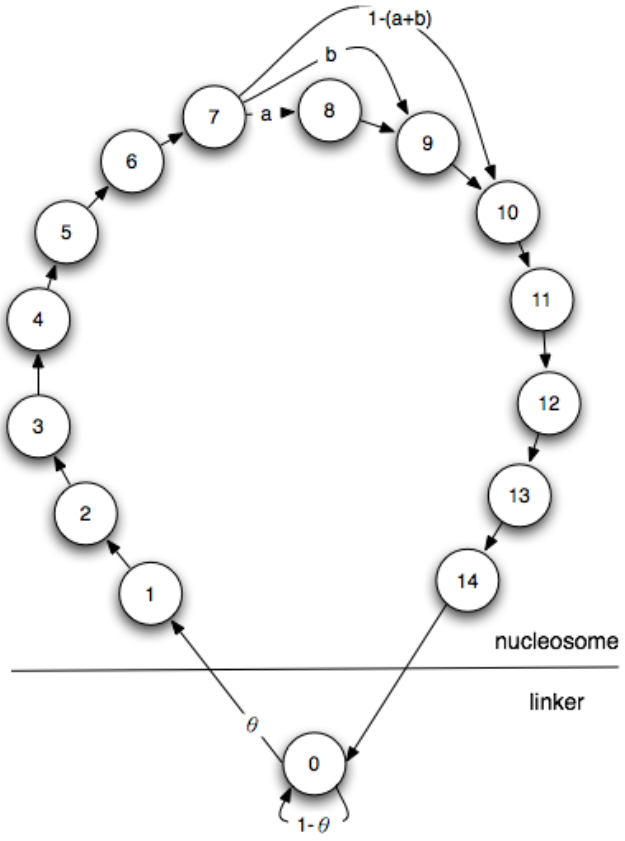


Figure 2.4: The states diagram of the S_i variables. Each node represents a possible assignment, and the edges represent the possible transitions. The label of each edge is the probability of this transition (1 if missing)

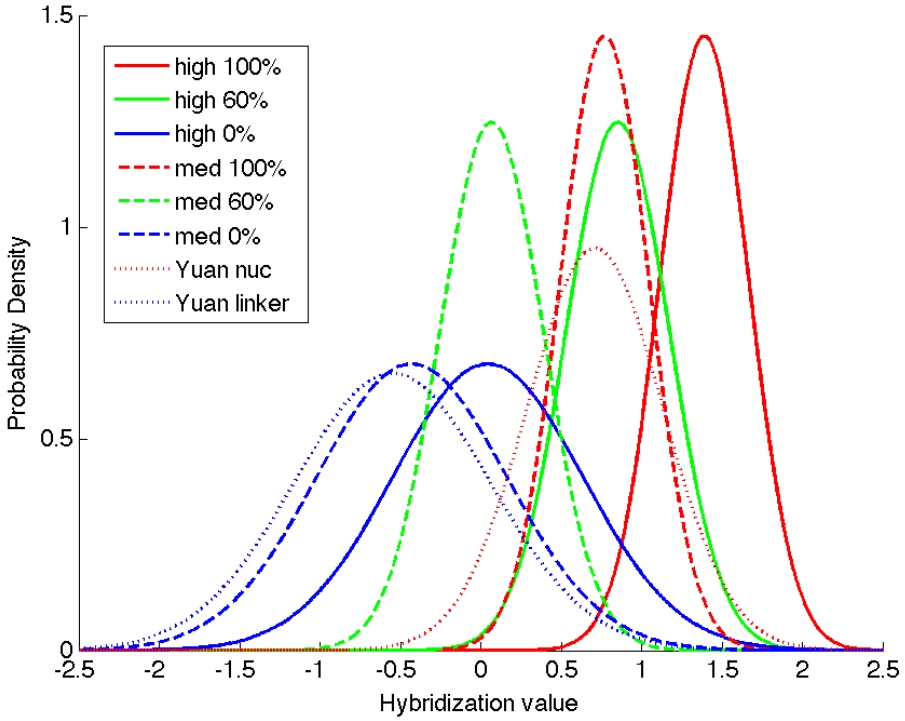


Figure 2.5: Some examples of the normal distribution density functions for $P(P_i \dim C_i, L_i)$. In red shown distributions for 100% coverages, in green for 60% and in blue for 0%. The highest occupancy level shown in a solid line, medium level in a dashed line, and the normal distribution used by Yuan et al. [2005] shown in a dotted line. When looking at the same level in our model, it is clear that the red gaussian is always to the right of the green one, which is to the right of the blue one. Moreover, we can see that our medium level is a bit higher than the single level used by Yuan et al. [2005].

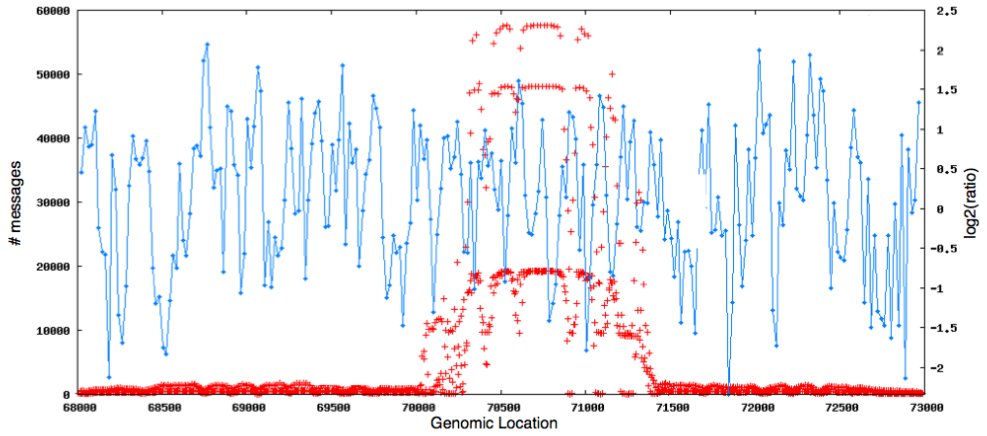


Figure 2.6: Number of messages passed in each genomic location. Probe values shown in blue, and number of messages in red. We see that in location 70,000-71,500 the loopy algorithm has a problem in converging, and the number of messages is rising

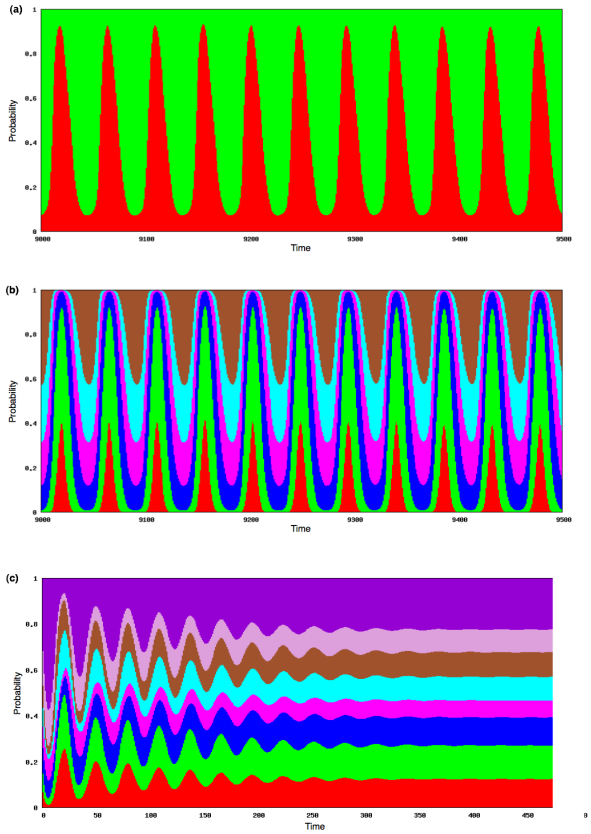


Figure 2.7: Messages content over time. **(a)** A message over a single S_i variable, stating whether there is a nucleosome in these 10 bp (green) or not (red). **(b)** A message over a single C_j variable stating what is the coverage of this probe (0% - red, 20% - green, 40% - blue, 60% - pink, 80% - cyan and 100% - brown). Both messages oscillate in a coordinated manner, such that when there is a nucleosome in (a), there are higher probabilities of high coverage in (b). **(c)** A converging message is shown here, just as a reference. We see the oscillation at the beginning, but as time advances it is converging to its final value.

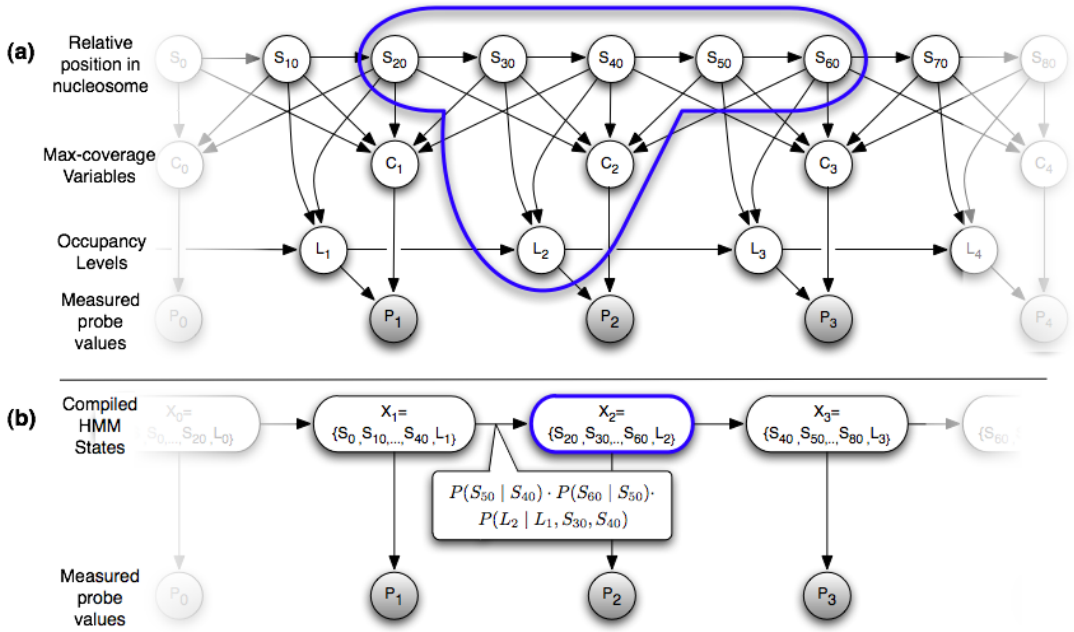


Figure 2.8: **(a)** graphical model as in Figure 2.3. All variables affecting the value of P_i are surrounded in blue. **(b)** Compilation of the model into an HMM: the states of X_i's denote the combination of S_j, C_i, L_i variables connected to the i^{th} probe. P_i are as in (a). The corresponding variable to the blue group in (a) is shown here in blue as well.

Chapter 3

Results

3.1 Small Scale Examples

We applied our model to genomic data of mid-log nucleosome positions in yeast from Yuan et al. [2005]. As described above, we used their nucleosome calls to train parameters for the model. In addition, we iteratively optimized the parameters of the model using standard EM algorithm, although this did not change the predictions significantly (data not shown).

By using the Viterbi algorithm, we find the MAP nucleosome organization given the probe measurements and use it to call nucleosomes. To validate the accuracy of our predictions we compared our nucleosome calls to the original calls by Yuan *et al.*, to a more recent set of predictions using higher-density arrays [Lee et al., 2007] (also during mid-log growth), and to a compiled set of experimentally verified nucleosome positions [Segal et al., 2006]. Figures 3.1, 3.2, 3.3 demonstrate the model predictions on several selected genomic regions. In addition to the most likely arrangement found by the Viterbi algorithm, we also plot the posterior probability of each location to be occupied by a nucleosome (calculated using the Forward-Backward algorithm). In Figure 3.1 we show the genomic region surrounding the promoter of the gene CHA1. We see different behaviors on both sides of the promoter: along the coding region of the upstream gene VAC17, our nucleosome calls match very nicely all tracks shown (literature, and other high-throuput methods). In contrast, downstream to the promoter our predictions are inconsistent with the positions reported in the literature. A closer look reveals that neither Yuan *et al.* nor Lee *et al.* calls succeed in predicting the literature calls on this area, which might suggest that the literature data is not accurate, or was tested under different conditions. In Figure 3.2 we show a genomic loci where Yuan *et al.* predicted "fuzzy" nucleosome locations. Al-

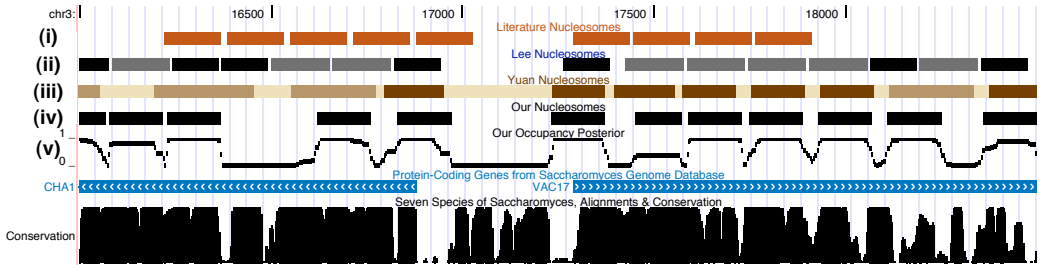


Figure 3.1: Example of our nucleosome calls compared to previous works displayed using the UCSC genome browser [Kent et al., 2002]. (i) Literature-based nucleosomes as curated by Segal et al. [2006]; (ii) Nucleosome calls from Lee et al. [2007]; (iii) Nucleosome calls from Yuan et al. [2005], where localized nucleosomes are dark brown, and fuzzy ones are light brown; (iv) Our nucleosome calls using MAP nucleosome positions; (v) The posterior probability of occupancy by a nucleosome according to our model. The CHA1 promoter (chromosome 3). In this region our calls match Yuan *et al.* and Lee *et al.*, and sometimes disagree with the literature locations.

though we use the same data, one can see how our algorithm has overcome this problem, and that our calls match very nicely those of the literature and of Lee *et al.* As these examples show, our method achieved high accuracy in calling nucleosome positions with regard to previous works, including both high- and low-throughput assays.

3.2 Genome Scale Validations

To further validate our results, we compared the different methods on a genomic scale. To calculate the agreement of two calling methods, we computed the match (sensitivity and specificity) of the calls. This was done in an unbiased way, by considering the limited genomic regions that were scanned by the two compared methods, and calculating the distances between center positions of predicted nucleosomes. If the centers of two nucleosomes were k bp or less apart, we say these nucleosomes match (see Figure 3.4 for details).

By changing the distance threshold that defines a match, we can explore different levels of accuracy. We used the calls of Lee *et al.* as a reference set to compare to the predictions by Yuan *et al.*'s and to our method. We divided the scanned genomic regions into two sets, those where Yuan *et al.* found localized nucleosomes, and those where they could only find fuzzy

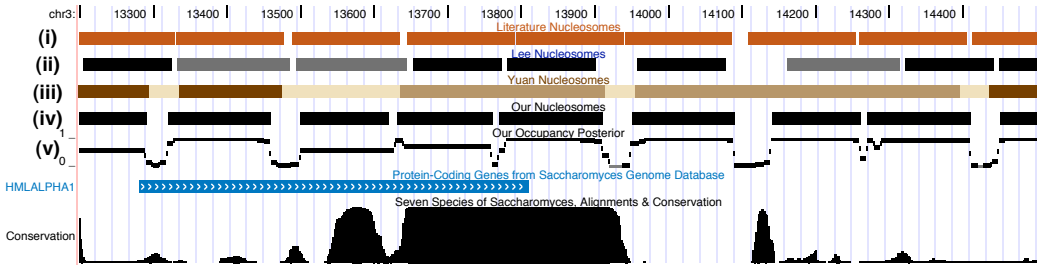


Figure 3.2: The HMLα1 gene (chromosome 3). (i)-(v) as in Figure 3.1. This region demonstrates the improvement over Yuan *et al.*'s calls, as we better explain the areas they have described as fuzzy nucleosomes. Moreover, our explanation of such fuzzy areas, matches that of Lee *et al.* and the literature positions.

nucleosomes, which means the data were not as conclusive. Figure 3.5 show the sensitivity and specificity in the regions where Yuan *et al.* had conclusive calls. We can see that in these regions our fully automated calls are as accurate as the hand curated nucleosome calls of Yuan *et al.* Once we look at the less conclusive data, where Yuan *et al.* found only fuzzy nucleosomes, our advantage is very clear (Figure 3.6). In these regions we succeed to predict many more of Lee *et al.*'s nucleosomes, while keeping our specificity very high (almost as high as in the localized genomic loci). In Figure 3.7 we compared all three nucleosome calls sets to the literature-based ones. In order to do so in the most unbiased way, we looked only at genomic loci scanned by all three methods. We then narrowed the literature-based set to these genomic locations (48 nucleosomes out of the 100 in the curated set), and compared our calls. As Figure 3.7 shows, our algorithm has somewhat better performance, even though Lee *et al.* used a higher resolution array.

To conclude, our results clearly show that our algorithm succeeded in exploiting the most out of the array. Not only we do better than Yuan *et al.* (which used the same data), but also when comparing to the literature-based set, our performance is comparable to that of Lee *et al.* who used a five-fold denser array.

3.3 Cell Cycle Results

After validating our results to the data used by Yuan et al. [2005], we want to apply our method on new data, and to learn about the nucleosome dynamics.

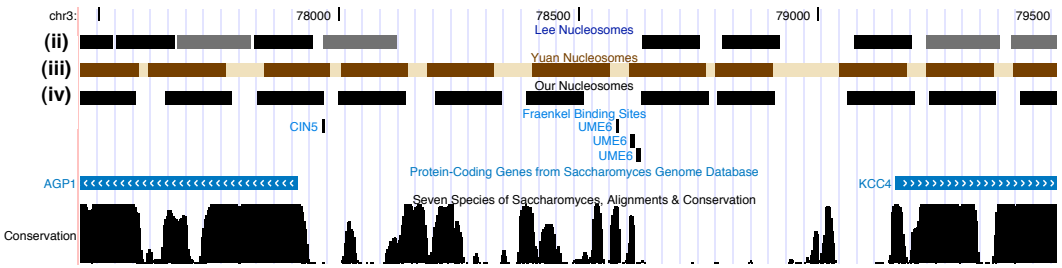


Figure 3.3: The AGP1 promoter (chromosome 3). (ii),(iii),(iv) as in Figure 3.1. To emphasize the significance of our higher-resolution calls, we add another track showing transcription factor binding sites, as reported by Harbison et al. [2004]. As we see, three binding sites of the transcription factor UME6 were found around position 78600. These sites match the known recognition sequence of UME6, and are also supported by a significant ChIP call ($p < 0.001$) [Harbison et al., 2004]. A closer look reveals that according to the calls of Yuan *et al.*, only one of these three binding sites is accessible (not covered by a nucleosome), but according to our calls all three binding sites are on linker DNA, hence available to the factor UME6.

We applied our algorithm to a set of time-series experiments (*i.e.*, nucleosome positions in cells advancing synchronously through the cell cycle), to explore the dynamic aspects of nucleosome positions. In collaboration with Oliver Rando's lab, we received the same type of data as before, only from synchronized cells, taken every 10 minutes.

At first, it seems like the data from different time points are very similar, as shown in Figure 3.8. In some locations it appears as if there is no change in nucleosome positions over time, and in other we expect to see a slight change in a few nucleosomes, especially in the regulatory elements.

Further examination of the cell cycle data reveals that generally, nucleosome locations do not tend to change significantly, and the dynamics seems to come from the nucleosome occupancy changes. I'm currently working on new data to validate this assumption, and from a first look it seems true. I can see a gradient in the nucleosome occupancy, decreasing in promoters toward expression of this gene, and increasing afterwards. Once again, this are very preliminary results, and I will continue this work in my PhD.

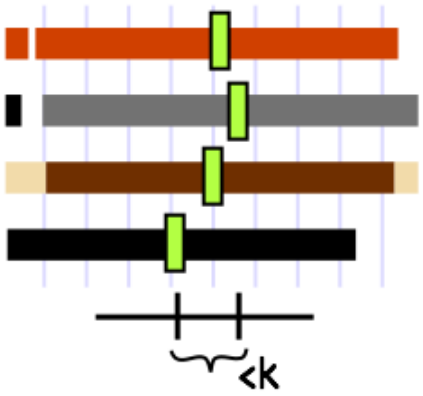
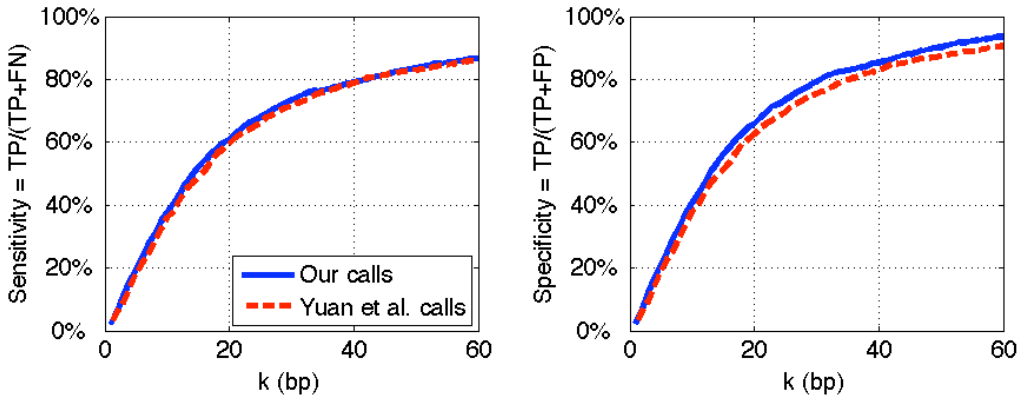
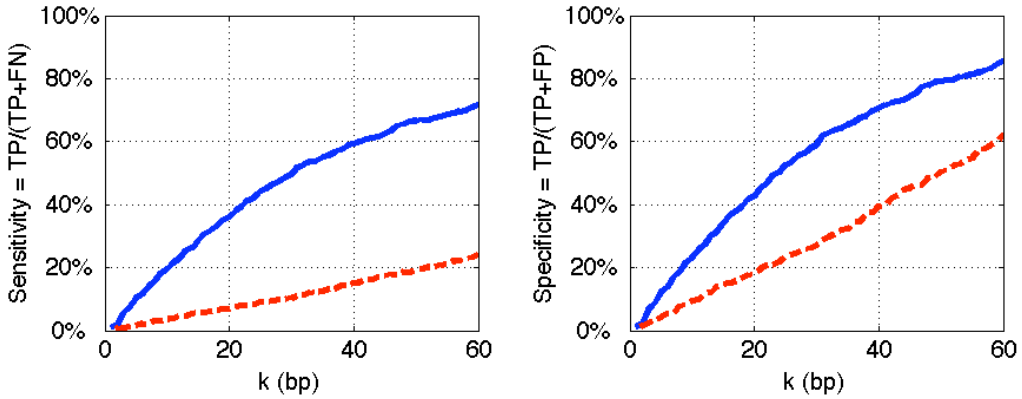


Figure 3.4: A measure to find matching nucleosomes. As in Figure 3.1, the blocks represent nucleosomes found by different methods. We calculate the center of each nucleosomes (shown in green), and if the centers of two nucleosomes differ by at most k bp, we conclude these nucleosomes match for this k.



(a) Sensitivity of localized nucs (b) Specificity of localized nucs

Figure 3.5: Comparison of our calls to other high-throughput calls. The sensitivity ($TP/(TP+FN)$) and specificity ($TP/(FP+TP)$), respectively, achieved for each distance threshold k, when comparing Yuan's and our calls to those of Lee *et al.* when examining regions where Yuan *et al.* found to be well localized. Our method shown in blue, and Yuan *et al.*'s shown in red.



(a) Sensitivity of fuzzy nucs (b) Specificity of fuzzy nucs

Figure 3.6: Comparison of our calls to other high-throughput calls. The sensitivity ($TP/(TP+FN)$) and specificity ($TP/(FP+TP)$), respectively, achieved for each distance threshold k, when comparing Yuan's and our calls to those of Lee *et al.* when examining regions where Yuan *et al.* predicted fuzzy nucleosome positions. Our method shown in blue, and Yuan *et al.*'s shown in red.

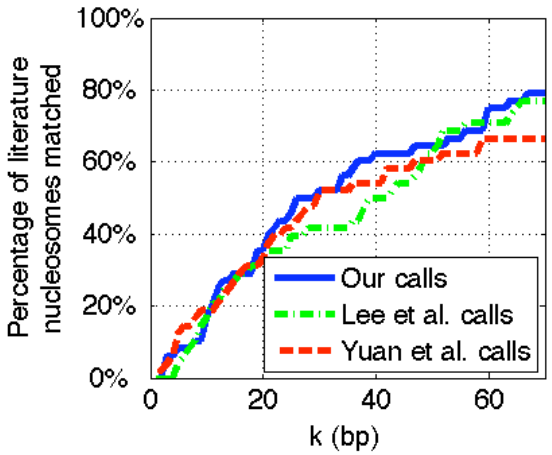


Figure 3.7: Comparison of all three high-throughput methods to a dataset of published nucleosomes [Segal et al., 2006].

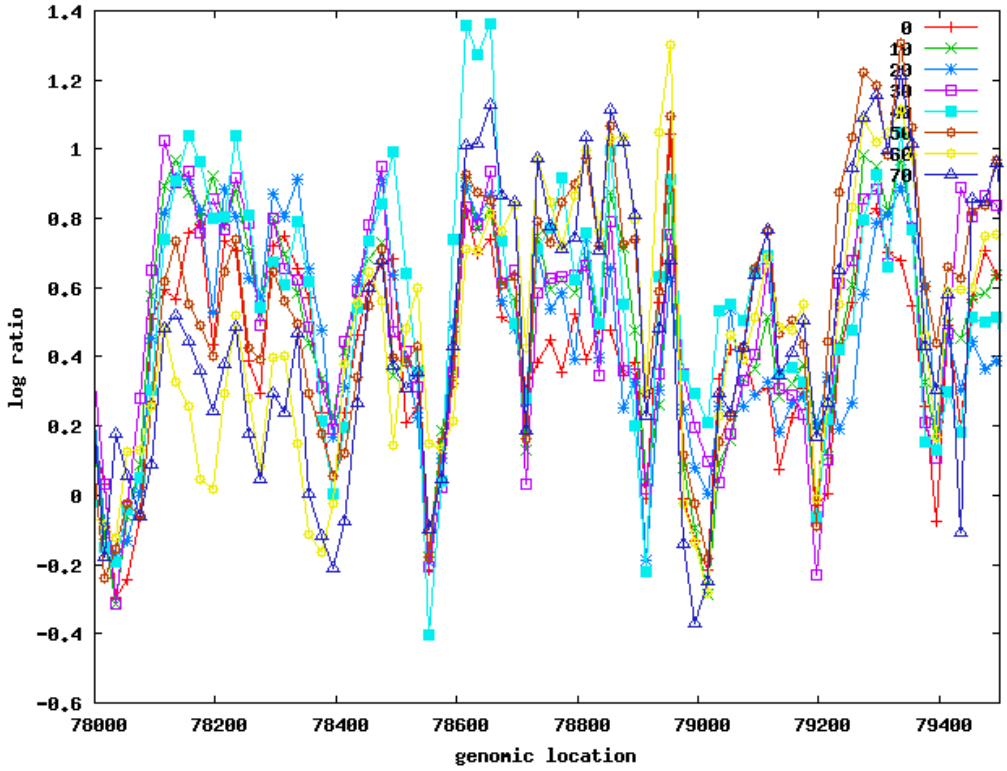


Figure 3.8: Raw data from synchronized cells, taken every 10 minutes. Shown are only 0-70 minutes due to space limitations. We can observe a high consensus linker and nucleosome at location 78550-78700. Another well positioned nucleosome can be found at location 79200-79400. On the other hand, at location 78700-78900 we do not see a distinct trend along the time points, and we can't make out the exact location of the nucleosomes.

Chapter 4

Discussion

In this work, we presented a fully automated computational method to analyze high-resolution microarrays measurements of nucleosomal occupancy along the genome. As opposed to previous methods, we showed how to extend the nucleosome calls' resolution beyond that of the measurements. This was done by designing a probabilistic graphical model which introduced a new dense layer of variables, and taking into account the predicted intensity of the signal in probes that are at the end of nucleosomes. We then showed how such a model can be compiled into a simple HMM, which enables a fast inference without any loss of accuracy. We applied this model to the genomic scale nucleosomal measurements of Yuan *et al.*, and predicted the nucleosome positions of thousands of nucleosomes.

As we showed, our method leads to better predictions from the same data, yielding 13% more nucleosomes than Yuan *et al.* (2660 compared to 2348). Furthermore, our calls were found to be about 20% more accurate, with regard to higher resolution microarrays (Lee et al. [2007]), and to published positions of nucleosomes. As shown in Figures 3.5, 3.6, these improvements were mainly obtained in problematic regions for Yuan *et al.*'s algorithm, where they could not specify the exact position of nucleosomes, and defined them as fuzzy. Moreover, this improvement was done in a fully automated manner, as opposed to the manual curation done by Yuan *et al.*

Although better calls can be obtained by the five fold denser arrays of Lee *et al.* or by new sequencing methods [Albert et al., 2007], we believe that our algorithm will be useful by making the most of the available measurements using the printed arrays of Yuan *et al.*, or similar ones [Liu, 2006]. The cost of such arrays is much lower than the alternative ones, and as we showed here their resolution is not dramatically different.

This higher accuracy achieved by our algorithm opens the way for a better understanding of the role nucleosomes play in transcriptional regulation.

When it comes to the position of nucleosomes in regulatory regions, every base pair counts. This is due to the typically short length of regulatory binding sites, and the tremendous role they play in transcriptional regulation. In this setting, a higher resolution of nucleosome calls will allow to separate the accessible sites from the unapproachable ones.

4.1 Future Work

In this work I focused on developing a fast, high-resolution method for locating nucleosomes along the genome. In my PhD work, I plan to apply this method to answer more elaborated questions to better understand the nucleosome positions pattern and dynamics. For example, I'm currently applying my algorithm to a set of time-series experiments (*i.e.*, nucleosome positions in cells advancing synchronously through the cell cycle, or cells responding to external stimuli), and explore the dynamic aspects of nucleosome positions. The method described here facilitates automatic and accurate nucleosome positioning from this wealth of data. Another interesting question can be asked on NFR-like elements. As we explained before the nucleosome free region is an area upstream to the transcription start site, which is usually unoccupied by nucleosomes. Currently, there is not much knowledge on such elements in higher organisms, but once data from these organisms will be available, I can apply our algorithm to it and answer this question.

On the computational aspect of this work, the compilation of the graphical model can be extended to more general models, and tested under different conditions.

Finally, I would also like to extend this model to handle the new ChIP-seq data, which I think will dominate this field soon.

Bibliography

I. Albert, T. N. Mavrich, L. P. Tomsho, J. Qi, S. J. Zanton, S. C. Schuster, and B. F. Pugh. Translational and rotational settings of H2A.Z nucleosomes across the Saccharomyces cerevisiae genome. *Nature*, 446(7135): 572–576, Mar 2007.

N. Friedman, D. Geiger, and N. Lotner. Likelihood computations using value abstraction. In *Proc. Sixteenth Conf. on Uncertainty in Artificial Intelligence (UAI)*. 2000.

C.T. Harbison, D.B. Gordon, T.I. Lee, N.J. Rinaldi, K.D. Macisaac, T.W. Danford, N.M. Hannett, J. Tagne, D.B. Reynolds, J. Yoo, E.G. Jennings, J. Zeitlinger, D.K. Pokholok, M. Kellis, P.A. Rolfe, K.T. Takusagawa, E.S. Lander, D.K. Gifford, E. Fraenkel, and R.A. Young. Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99–104, 2004.

W. J. Kent, C. W. Sugnet, T. S. Furey, K. M. Roskin, T. H. Pringle, A. M. Zahler, and D. Haussler. The human genome browser at UCSC. *Genome Res*, 12(6):996–1006, Jun 2002.

D. S. Latchman. *Gene Regulation: A Eukaryotic Perspective*. Taylor & Francis, 5 edition, 2005.

C. K. Lee, Y. Shibata, B. Rao, B. D. Strahl, and J. D. Lieb. Evidence for nucleosome depletion at active regulatory regions genome-wide. *Nat Genet*, 36(8):900–905, Aug 2004.

W. Lee, D. Tillo, N. Bray, R. H. Morse, R. W. Davis, T. R. Hughes, and C. Nislow. A high-resolution atlas of nucleosome occupancy in yeast. *Nat Genet*, 39(10):1235–1244, Oct 2007.

C. L. Liu. Dynamic high-resolution mapping of histone tail modifications in s. cerevisiae. *Ph.D. thesis, Harvard University*, 2006.

K. Luger, A. W. Mader, R. K. Richmond, D. F. Sargent, and T. J. Richmond. Crystal structure of the nucleosome core particle at 2.8 A resolution. *Nature*, 389(6648):251–260, Sep 1997.

K. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in AI (UAI)*, 1999.

D. K. Pokholok, C. T. Harbison, S. Levine, M. Cole, N. M. Hannett, T. I. Lee, G. W. Bell, K. Walker, P. A. Rolfe, E. Herbolsheimer, J. Zeitlinger, F. Lewitter, D. K. Gifford, and R. A. Young. Genome-wide map of nucleosome acetylation and methylation in yeast. *Cell*, 122(4):517–527, Aug 2005.

L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, pages 4–16, 1986.

H. Reinke, P.D. Gregory, and W. Hrz. A transient histone hyperacetylation signal marks nucleosomes for remodeling at the PHO8 promoter in vivo. *Mol. Cell*, 7:529–538, Mar 2001.

E. Segal, Y. Fondufe-Mittendorf, L. Chen, A. Thastrom, Y. Field, I. K. Moore, J. P. Wang, and J. Widom. A genomic code for nucleosome positioning. *Nature*, 442(7104):772–778, Aug 2006.

U. Venter, J. Svaren, J. Schmitz, A. Schmid, and W. Hrz. A nucleosome precludes binding of the transcription factor Pho4 in vivo to a critical target site in the PHO5 promoter. *EMBO J.*, 13:4848–4855, Oct 1994.

Y. Weiss, C. Yanover, and T. Meltzer. Map estimation, linear programming and belief propagation with convex free energies. In *Proceedings of the Twenty-Third Annual Conference on Uncertainty in AI (UAI)*, 1999.

G. C. Yuan, Y. J. Liu, M. F. Dion, M. D. Slack, L. F. Wu, S. J. Altschuler, and O. J. Rando. Genome-scale identification of nucleosome positions in S. cerevisiae. *Science*, 309(5734):626–630, Jul 2005.