Machine learning approach to the study of chromatin

Pol Cuscó Pons

TESI DOCTORAL UPF / 2017

DIRECTOR DE LA TESI

Dr. Guillaume Filion,

GENOME ARCHITECTURE
GENE REGULATION, STEM CELLS AND CANCER
CENTRE FOR GENOMIC REGULATION

ABSTRACT

Since the appearance of high throughput sequencing technologies, biological data sets have become increasingly large and complex, which renders them practically impossible to interpret directly by a human. The machine learning paradigm allows a systematic analysis of relationships and patterns within data sets, making possible to extract information by leveraging the sheer amount of data available.

However, violations of basic machine learning principles may lead to overly optimistic estimates, a prevalent problem known as overfitting. In the field of protein folding, we found examples of this in published models that claimed high predictive power, but that performed poorly on new data.

A different problem arises in epigenetics. Issues such as lack of reproducibility, heterogeneous quality and conflicts between replicates become evident when comparing ChIP-seq data sets. To overcome this limitations we developed Zerone, a machine learning-based ChIP-seq discretizer capable of merging information from several experimental replicates and automatically identifying low quality or irreproducible data.

RESUM

Des de l'aparició de les tecnologies de seqüenciació d'alt rendiment, els conjunts de dades biològiques han esdevingut cada cop més grans i complexes, la qual cosa els fa pràcticament impossibles d'interpretar manualment. El paradigma de l'aprenentatge automàtic permet fer una anàlisi sistemàtica de les relacions i patrons existents en els conjuts de dades, tot aprofitant l'enorme volum de dades disponibles.

No obstant això, una aplicació poc curosa dels principis bàsics de l'aprenentatge automàtic pot conduir a estimacions massa optimistes, un problema prevalent conegut com a sobreajust. En el camp del plegament de proteïnes, en vam trobar exemples en models publicats que afirmaven tenir un alt poder predictiu, però que es comportaven de forma mediocre devant de dades noves.

En el camp de l'epigenètica, problemes com la falta de reproducibilitat, qualitat heterogènia i conflictes entre replicats esdevenen evidents quan es comparen diferents conjunts de dades de ChIP-seq. Per superar aquestes limitacions vam desenvolupar Zerone, un discretitzador de ChIP-seq basat en aprenentatge automàtic que és capaç de combinar informació de diferents replicats experimentals i d'identificar automàticament dades de baixa qualitat o irreproduïbles.

TABLE OF CONTENTS

Abstract 3
Introduction 11
- About machine learning 13
- Support vector machines 15
- The generalization problem 19
- Hidden Markov models 22
- About chromatin 27
- Chromatin immunoprecipitation techniques 31
- Working with ChIP-seq data 34

Machine learning: how much does it tell about protein folding rates?
- Abstract 43
- Introduction 44
- Results 45
- Discussion 58
- Methods 59
- References 61

Zerone: A ChIP-seq discretizer for multiple replicates with built-in quality control
- Abstract 69
- Introduction 70
- Methods 73
- Results 83
- Discussion and conclusions 92
- References 96

Discussion 101
Conclusions 105
References 107
Annex 1 113
Annex 2 123

CHAPTER 1

Introduction

About machine learning

In the last years, the amount of biomedical data available to the scientific community has grown enormously, partly due to the appearance of high-throughput sequencing technologies. Analysis of these complex sets of information is challenging but well suited for a family of computational methods collectively known as machine learning.

In machine learning, a computer program learns to perform a task without being explicitly programmed how to solve the task at hand. It can be said that a machine learns when its performance on a task increases with experience, that is with increased exposure to representative data. Both the definitions of performance and task depend on the particular problem being treated. There are many types of problems that can be approached with machine learning, and many different strategies have been devised. In general, they can be divided into three groups: supervised learning, unsupervised learning, and reinforcement learning strategies. In all cases, each input datum is defined by an arbitrary number of features that quantify different aspects of it. For instance, two features that could define a genomic sequence would be its length in nucleotides and its GC-content.

In supervised learning, after being exposed to enough training examples, the algorithm models the relationship between the input features and a known output variable. This model is used to predict the unknown output variable of new, previously unseen examples. In unsupervised learning the output variables of the training

examples are not available (or not used), and the goal is to assign them to the data set, thus labeling each datum. This usually implies clustering the training examples into groups of similar objects, while maintaining the different objects in separate groups. The number of groups is not an intrinsic property of the data, but more a decision made by the analyst, who usually must find a balance between intra-group homogeneity and model complexity (Akaike 1974; Schwarz 1978). The class labels that are learned with an unsupervised strategy can be used downstream in a supervised setting. The last strategy, reinforcement learning, involves maximizing the reward that an agent receives after performing some actions on an environment in response to stimuli, however its review is outside the scope of this dissertation.

There are two main types of tasks in machine learning: regression and classification. In regression problems the aim is to approximate the output of a real-valued function as accurately as possible. For instance, one may want to predict the folding rate of a given protein after having trained a model with examples of other proteins whose folding rates are known, in a supervised fashion (see Chapter 2). This is usually accomplished by fitting a parameter vector θ, that constitutes the model to be learned, and minimizing a cost function $J(\theta)$ on the training set.

$$J(\theta) = \frac{\sum_{i=1}^{m} (\theta \cdot \mathbf{x_i} - y_i)^2}{2m},$$

where m is the number of training examples, $\mathbf{x_i}$ is the feature vector of the ith training example and y_i is the output scalar variable that corresponds to the same training example. This means that the algorithm will learn a model θ to approximate y_i based on $\mathbf{x_i}$. The parameter vector θ can be randomly initialized and updated iteratively using an optimization method such as gradient descent. Ideally, with noise-free data there will be no difference between $\theta \cdot \mathbf{x_i}$ and y_i at the end of the training, and the model can be then used to predict the unknown output value of a new, unseen example. The result is equivalent to performing linear regression, therefore it is assumed that the output variable is a linear combination of the feature vector. Because derivative-based optimization methods (such as the early mentioned gradient descent) require a convex cost function to converge to a unique global minimum, the error of the predicted to actual output values is squared. It is precisely this error as measured by the cost function $J(\theta)$ what is minimized. The $1/2$ constant simplifies the derivative expression.

Support vector machines

The other type of problem is classification, in which the algorithm assigns discrete labels to objects and determines to what class they belong. Classification problems can be solved using a supervised strategy. Different methods exist, but the focus of this dissertation is on a type of binary classifier called support vector machine (see Chapter 3). This approach consists in finding the hyperplane $\mathbf{w}$ that separates the two classes of training examples

while at the same time maximizing the margin M between such hyperplane and the closest points in the data set (Fig. 1).

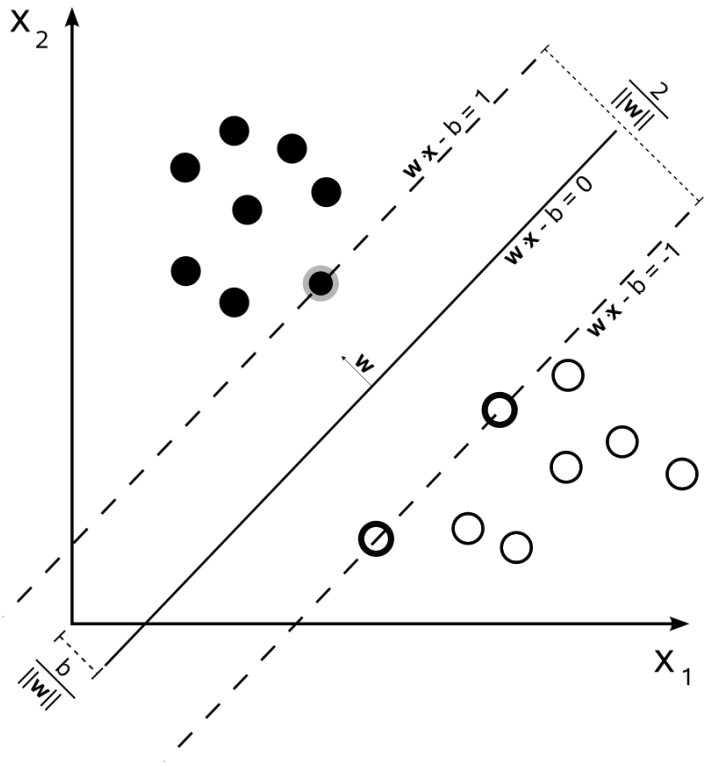

Figure 1. Support vector machine classification. The two classes of data, depicted as black and white circles and defined by the arbitrary features $\mathbf{x_1}$ and $\mathbf{x_2}$ in the example, are completely separated by the hyperplane that maximizes the margin. Other approaches that do not attempt to maximize the classification margin may separate the two populations completely in the training data set, but will tend to generalize poorly when presented with new data. Maximizing the margin increases the chances that new data points from each class will not be found on the wrong side of the decision boundary. The circled points, the ones closest to the hyperplane $\mathbf{w}$, are called support vectors and determine the position of such boundary.

The problem then is to maximize

$$M = \frac{2}{||\mathbf{w}||}$$

subject to the constraint

$$y_i(\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1$$

Where $y_i \in \{-1, 1\}$ is the label of the ith training example and b is an intercept term. This type of classification is called hard margin support vector machine and it is only possible if the two classes are linearly separable. If there is any degree of overlap between the two classes that would require to fit a nonlinear classifier, two solutions are available: using a soft margin classifier, in which the constraint is relaxed to allow some points to be misclassified at a specified cost; and/or mapping the features to a higher-dimensional space in which they are linearly separable. The latter can be accomplished by using the so called kernel trick, that is using a kernel function Φ to compute the similarity between all pairs of points in the original feature space (Fig. 2).

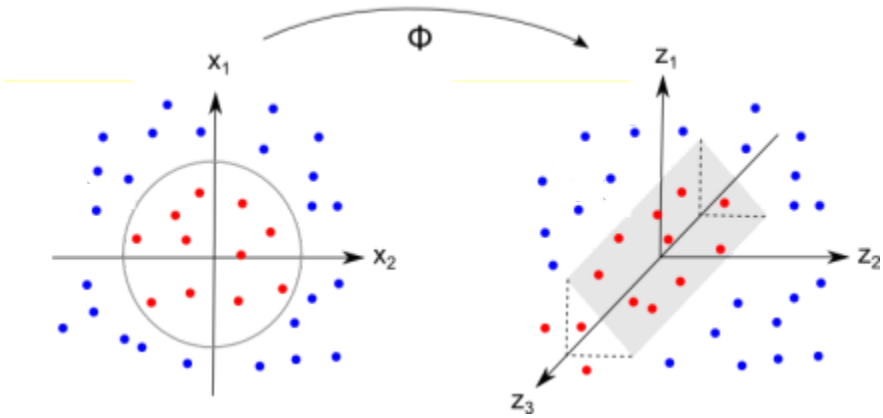

Figure 2. Kernel trick. A kernel function maps the data points from the original, low-dimensional feature space to a new, higher-dimensional kernel space. Generating a new set of coordinates by computing the similarities of all pairs of points ensures that every point is linearly separable from any other. For convenience, a mock 3-dimensional space is depicted on the right, when in reality the kernel space could have as many dimensions as data points, representing the proximity of each point to every other.

The most widely used kernel is the Gaussian radial basis function kernel, that computes the proximity between two points as:

$$\Phi(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\sigma^2}\right)$$

Where $\mathbf{x}$ and $\mathbf{x}'$ are the two points in the data set which proximity is measured, and σ is a parameter that controls the decay of the exponential function. Nonetheless, the kernel function is usually parameterized by $\gamma = \frac{1}{2\sigma^2}$, which gives the simpler form

$\Phi(\mathbf{x}, \mathbf{x}') = \exp(-\gamma||\mathbf{x} - \mathbf{x}'||^2)$. The kernelized features can be then fed to the support vector machine to learn a classification.

The generalization problem

All machine learning techniques learn a model or representation of the data that is used for training. In the case of supervised learning, the quality of this model can be evaluated by measuring the cost J at the end of the training, but this does not guarantee that, once confronted to new data, the model will be able to make accurate predictions. If this happens, it means that the model has learned to characterize particular training examples that were not representative of the underlying structure of the data but outliers caused by random noise or biased sampling. This situation is called overfitting and can be diagnosed if the cost on the unobserved data set is significantly greater than the cost on the training set (Fig. 3). Complex models are more prone to overfitting (usually the complexity of the model is determined by the number of parameters to fit, that is the dimension of θ). Simpler models on the other hand may be underfit: they are not able to represent certain structures present in the data. An example of the latter would be to fit a linear function to the data set in Fig. 3.

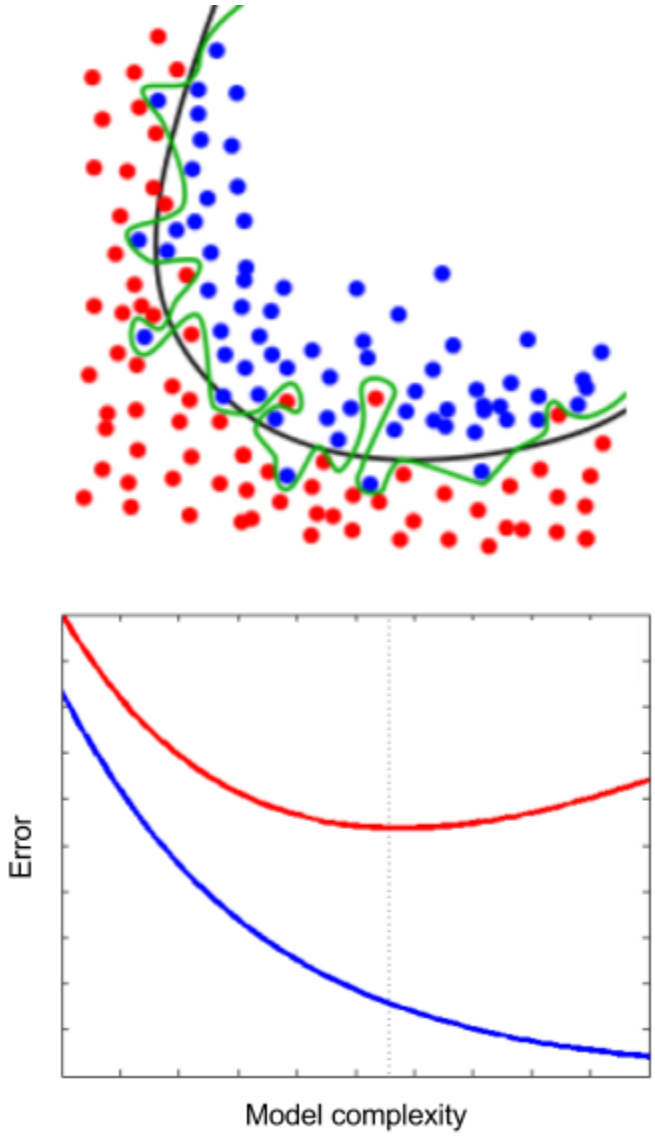

Figure 3. Generalization and overfitting. The upper panel shows a data set composed of two classes, shown as red and blue dots in an arbitrary feature space. The generalizable model (black line) fails to classify a few points, but captures the underlying structure of the data set. On the other hand, the overfit model (green line), even when classifying correctly all of the training points, has learned a nonrepresentative structure of the data and will likely perform poorly on new, unseen examples. The lower panel shows the classification error of models with different complexity on

training (blue) and new (red) data. As the aim is to minimize the generalization error, the best model would be the one that lies on the black dotted line (this would correspond to the black model on the upper panel). More complex models fit better the training data but do not generalize well, they are overfit (green model on the upper panel).

Poor generalization can also arise when the training set contains insufficient examples, which results in undersampling of the feature space. Also, complex models require to learn from bigger data sets. This is related to the fact that, as the dimensionality of the model increases, the amount of data available becomes more sparse in higher dimensional feature spaces, which leads again to undersampling (Hughes 1968). For these reasons, the fit of a model can also be evaluated as a function of the data set size, which may help diagnose if the current model is properly fit, if it is too simple, of if the amount of available data is insufficient for the complexity of the model. This questions are treated in more detail in Chapter 2.

A common approach to evaluate the quality of the models is to perform a cross validation. This process consists in partitioning all available data into two groups, then one of them is used to train the model and the other one is used to make predictions and compare them with the known values of the output variable. Since the quality of the model depends on the amount of data used in the training, one wants to use as many examples as possible to train. The extreme case of this approach is called leave-one-out cross validation, where all m examples but one are used in training, and the model is evaluated in the remaining one. The process is repeated m times, each time testing on a different example, and

the error or cost is averaged over all examples. Because this is usually very time-consuming to the point of being impractical, analysts use scaled-down version of this strategy called k-fold cross validation. Specifically, the data set can be partitioned into two groups of uneven size, one of size $(k-1)m/k$ for training, and one of size m/k for testing. The process is repeated k times, and the results are averaged. The generalization error in this case is not estimated as accurately as in leave-one-out cross validation, but is usually good enough. A typical value for k is around 10, but the choice is up to the analyst and to the need for accuracy versus speed.

Hidden Markov models

In the unsupervised learning category the goal is to learn new representations of the input data that are based on some structure present in them. Specifically, the most common task is to segregate the data into homogeneous groups. In the case of genomic data, information is encoded along the length of the genome, so it makes sense to assume that there exists some kind of dependency between neighboring regions. Hidden Markov models assume that the system being modeled is a Markov process, that is a sequence of steps where at each step the system is in a particular state, and where the current state stochastically depends only on the state at the previous step (Fig. 4). For instance, every genomic window can be considered either enriched in a certain chromatin mark or not, and this enrichment state depends only on the state of the previous window (see Chapter 3). In a hidden Markov model, the states are not directly observable, they are hidden, and the only way to

estimate them is indirectly through a series of observations. At each step, the system makes an emission, which is the observable value, and this emission depends again stochastically only on the current state. Following the previous example, each window has an associated read count that depends on the hidden state of the window (Fig. 4). In a hidden Markov model with discrete emissions, the emission probabilities of each state determine which of a finite number of emissions will be observed, in a model with continuous emissions, the emission parameters define the probability distribution from which the emissions are sampled in each state. A model is thus defined by the state transition probabilities and by the emission probabilities.

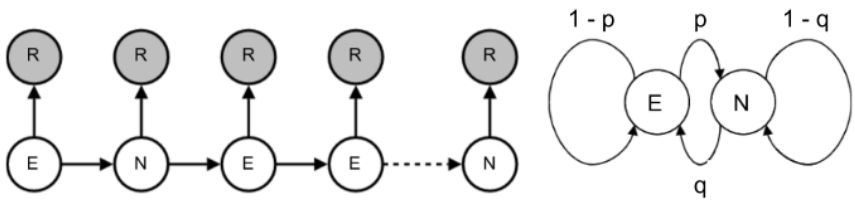

Figure 4. Example of hidden Markov model on the genome. The diagram on the left shows a sequence of genomic windows (white circles) each in a state of enrichment (E) or not (N). Each window has an associated observable read count (R in grey circles). The state of the current window depends stochastically on the state of the previous one. The diagram on the right represents the probabilities (p, q and their complementaries) of transitioning from one state to the next.

Three different problems can be solved by hidden Markov models (Rabiner 1989) (see Appendix 2), depending on what data are available at the moment. The first is the evaluation problem, in which the aim is to compute the probability of a sequence of

observations given a known model. The forward-backward algorithm, that recursively computes the probability of the sequence from the first up to the current step, and from the next to the last, is used to solve this problem. The second is the decoding problem, that is to determine the most likely sequence of states given the sequence of observations and a known model. The Viterbi algorithm (Viterbi 1967) recursively computes the probability of a step being in a particular state, and selects the state with the highest probability at the current step, based on the state selected in the step before. Finally, there is the learning problem, in which a model is estimated given only the sequence of observations. The probabilities computed with the forward-backward algorithm are used by the Baum-Welch algorithm (Baum and Petrie 1966) to reestimate the model, that is the state transition and emission probabilities, starting from a random model and iteratively approximating another one with a greater likelihood. A detailed explanation of the different algorithms can be found in Appendix 2.

Starting with a sequence of observations, one can randomly initialize a model, then use the Baum-Welch (which internally uses the forward-backward) algorithm to find the maximum likelihood model, and then use the Viterbi algorithm with this learned model and the observations to determine the most likely state sequence. An unsupervised clustering of all steps (genomic windows in the example) can be done in this way. This approach takes into account not only the similarities between observations, but also their relative position along the sequence, when performing the clustering. The labels learned during the process can be later used to learn supervised predictive models with other data, for instance

to predict the level of gene expression of a particular genomic window given its enriched or unenriched state in a certain chromatin mark.

About chromatin

During the late XIX century, early observations of eukaryotic cells revealed that their nuclei were composed of a substance that Walther Flemming, in 1879, named chromatin, after its property of being stained by basophilic dyes. The composition of chromatin was partially determined a few years later, in 1881, when the botanist Eduard Zacharias showed that nuclein, the strange substance that Friedrich Miescher had discovered a decade before in the nuclei of leukocytes, was a major component of the chromosomes (reviewed in Dahm 2005). Later, in 1928, another important contribution was made when Emil Heitz observed the longitudinal differentiation of mitotic chromosomes and coined the terms euchromatin and heterochromatin (reviewed in Zacharias 1995).

Now we know that chromatin consists mostly of DNA and proteins, the most abundant of these proteins being the histones. Histones H2A, H2B, H3 and H4, the so called core histones, form an octamer made of two monomers of each, around which a stretch of about 147 base pairs of DNA wraps to form the nucleosome core (Fig. 5). The DNA strand is coiled 1.67 times around these histones before “leaving” the nucleosome core, then it extends freely as linker DNA for several dozen base pairs, and coils itself again around the next nucleosome. This pattern repeats itself throughout the genome, forming the “beads-on-a-string” structure (Fig. 6).

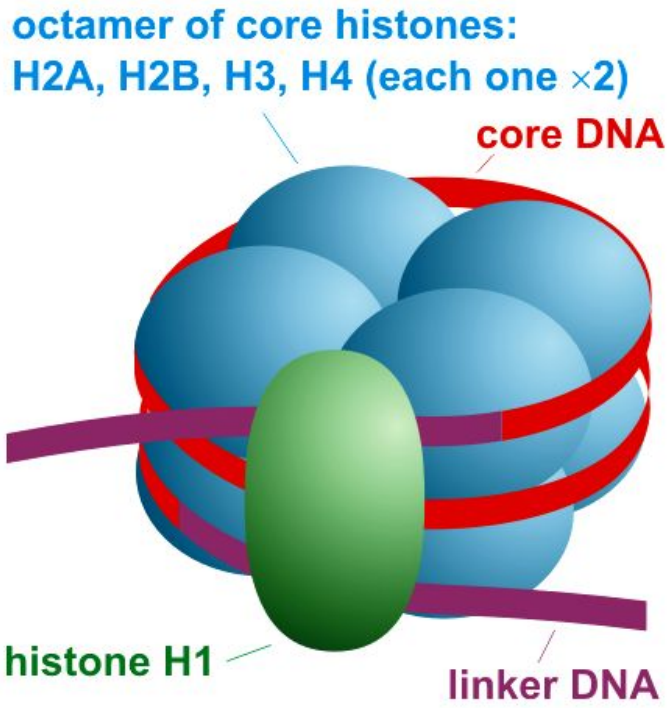

Figure 5. Nucleosome structure. An octamer of two units of each of the four core histones is wrapped in DNA to form the nucleosome core. The linker histone H1 stabilizes the complex and allows for higher order structure to form. Image from (Stryer 1995).

The linker histone H1 attaches to the nucleosome dyad, the place where the entering and exiting ends of the nucleosomal DNA meet, and stabilizes the nucleosome. Moreover, histone H1 facilitates the binding of consecutive nucleosomes which creates higher level, more compact structures (Medrano-Fernández and Barco 2016), like the classical 30 nm chromatin fiber found *in vitro* (Finch and Klug 1976), or the heterogeneous clutches of nucleosomes found in actual cell nuclei (Ricci et al. 2015).

Other scaffolding proteins compact the chromatin in successively higher order structures. In this way, the almost 2 m of DNA molecules contained in each human cell can be packaged inside a micrometer-order nucleus. The compacting function of chromatin proteins is especially evident during mitosis and meiosis, when chromosome condensation takes place and the archetypical X-shaped metaphasic chromosomes can be observed (Fig. 6).

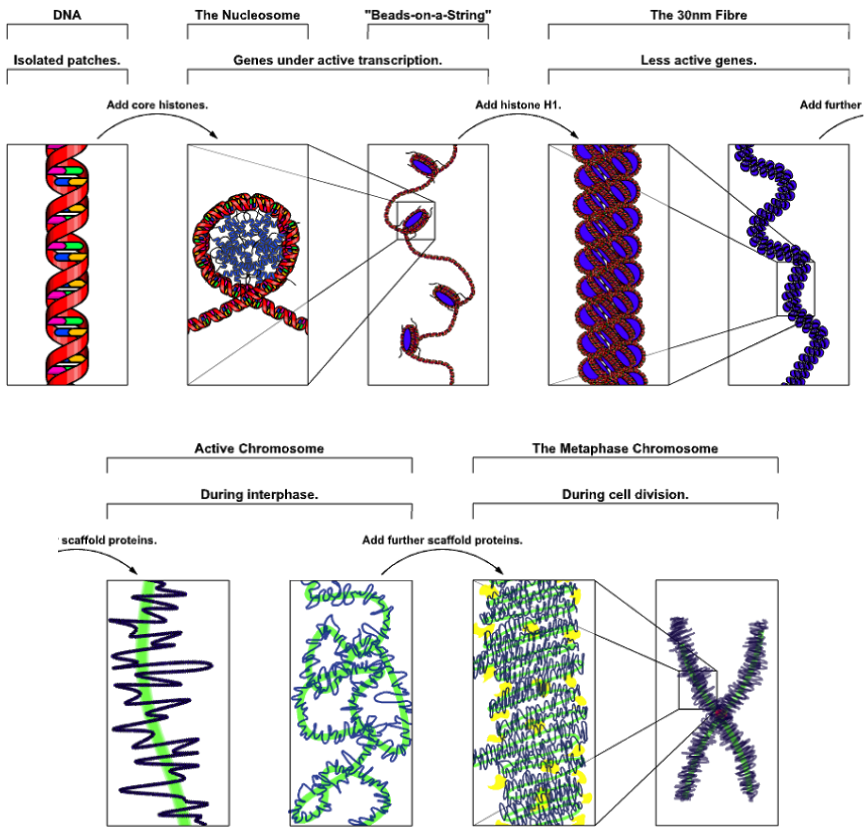

Figure 6. The different levels of chromatin compaction. From the naked DNA to the metaphasic chromosome, many layers of compaction are stacked, prominently the nucleosome, the "beads-on-a-string" structure and the 30 nm fiber.

Apart from its structural functions, chromatin plays an important role in gene regulation. The classical view states that chromatin comes in two forms, euchromatin and heterochromatin. Euchromatin comprises most of the human genome (International Human Genome Sequencing Consortium 2004), is relatively uncompacted and is associated with active transcription. In contrast, heterochromatin shows a higher degree of compaction and the genes it contains tend to be silenced through a variety of mechanisms. Originally, these two variants of chromatin were distinguished cytologically by staining, the latter acquiring a more intense color than the former. More recently, different authors have proposed alternative classifications based on genome-wide chromatin composition data that show different levels of expression and are enriched in different types of genes or other genomic features (Ernst and Kellis 2010; Filion et al. 2010). These chromatin states further subdivide the traditional binary nature of chromatin and explain better the compositional and functional variation observed inside each of the euchromatic and heterochromatic compartments.

Knowledge about the composition of this protein and DNA complex, and how it changes during the different phases of the cell cycle, during cellular differentiation, and in response to diverse stimuli, is thus critical to understand cell function as well as disease. In this regard, the scientific community has invested great effort in elucidating the nature of chromatin in multiple cell lines and conditions. Large consortia such as ENCODE have produced and released huge amounts of data that can be analyzed by independent researchers. Most of the data sets have been

obtained by means of specific immunoprecipitation (Brdlik et al. 2014), though alternate techniques such as DamID (van Steensel and Henikoff 2000), that does not make use of antibodies, have also been employed.

Chromatin immunoprecipitation techniques

Chromatin immunoprecipitation (ChIP) is an experimental technique developed to interrogate whether a certain DNA region is bound by a certain protein of interest (Solomon and Varshavsky 1985). It is the basis from which the other techniques discussed below stem, as they all use the same principle. The procedure starts by treating a cell culture or tissue with a reversible cross-linking agent like formaldehyde (Jackson and Vaughn 1978) or UV light (Gilmour and Lis 1985) in order to stabilize the bonds between the DNA and the proteins in the nucleus. After this, the cells are lysed and the chromatin is first isolated and then fragmented by means of sonication or enzymatic digestion. This fragments are subsequently immunoprecipitated with an antibody against the protein of interest and the DNA is then purified. The result is a DNA sample enriched for the sequences that were originally bound by the protein of interest. A particular DNA region may be further amplified by PCR and sequenced using a variety of methods to assess its relationship to the protein of interest. This procedure has a limited throughput as only one DNA region can be interrogated at once.

Genome-wide epigenomic analyses became common as scaled-up versions of the experiment were developed. The first to come to

light was a combination of ChIP and microarray technologies. The so called ChIP-on-chip uses a microarray with DNA probes that sample a large portion of the genome (Ren et al. 2000). The experiment is conducted as a ChIP but, at the end of the immunoprecipitation, the DNA is denatured and labeled with a fluorescent tag, and subsequently hybridized to the oligonucleotide probes in the microarray. Afterwards, as in other microarray-based techniques, data normalization, noise reduction and statistical analysis are required to call the significant regions of enrichment, which raises a challenge at the computational level.

Overall, ChIP-on-chip proved to be relatively expensive and labor-intensive, its coverage is limited by the number of probes present on the microarray, and the lack of standardization between different microarray platforms leads to further complications. Also, it is susceptible to several artifacts that cause a high background signal that may lead to false positive results (Waldminghaus and Kirsten 2010). For these reasons, with the advent of high-throughput sequencing technologies, ChIP-on-chip was mostly abandoned in favor of chromatin immunoprecipitation followed by high-throughput sequencing (ChIP-seq) (Brdlik et al. 2014). This approach allows for a truly genome-wide interrogation of the binding sites of chromatin proteins, with the only exception of repeated or other low complexity regions, whose state remains elusive due to their low mapping confidence.

The name of the technique is self-explanatory. After the immunoprecipitation part of the ChIP procedure, the DNA sample is sequenced in a high-throughput manner, the resulting reads are

aligned to a reference genome and their abundance is taken as a proxy for protein binding at every mappable location of the genome (Fig. 7).

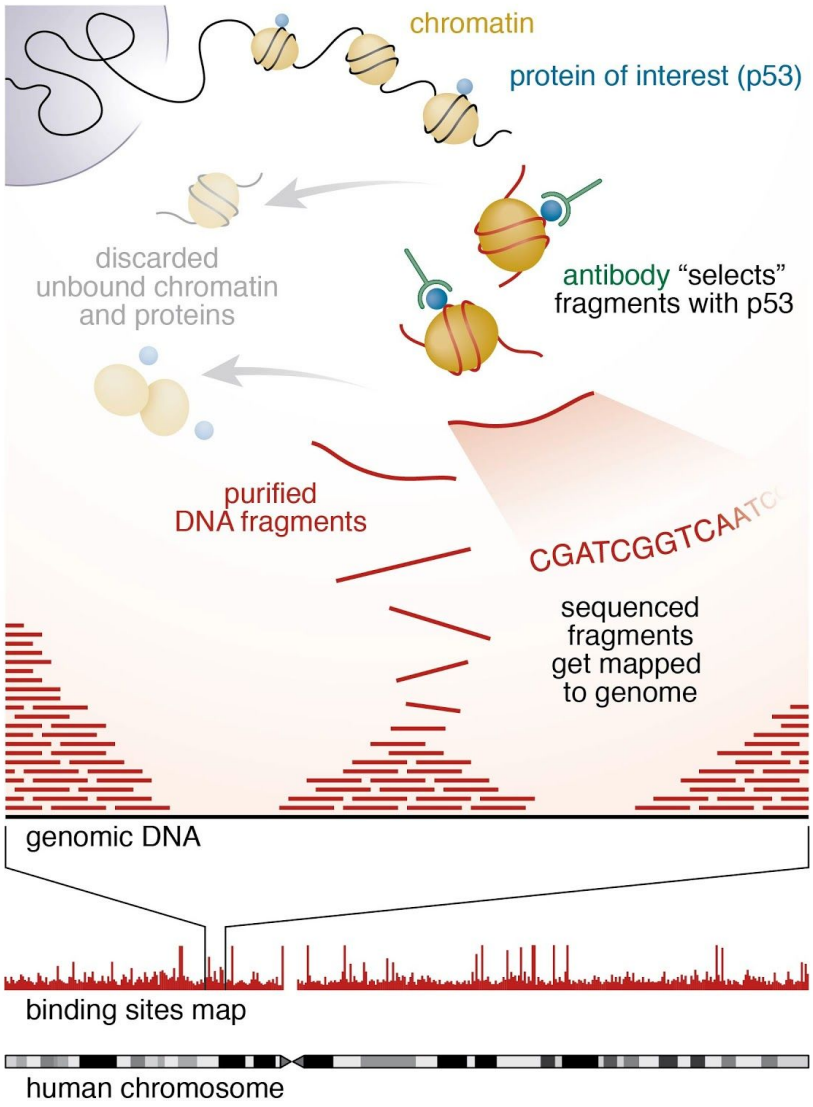

Figure 7. ChIP-seq procedure. Protein-DNA interactions are stabilized by means of cross-linking, chromatin is fragmented, the protein of interest (p53 in the example) is immunoprecipitated together with the DNA region it was binding, DNA is purified and sequenced, and finally the sequencing reads are aligned to a reference genome. Image from (BNL Newsroom).

It is good practice to generate a negative control sample for ChIP-seq experiments. To do so, the same procedure is used throughout but the immunoprecipitation step, that is the addition of the antibody specific against the protein of interest, is either omitted or performed with a non-specific antibody. Ideally this allows to identify locations that show a systematically biased read count. Also, experimental replicates help to elucidate what local enrichments represent a true protein binding location in contrast to a random increase in read count caused by any source of noise.

Working with ChIP-seq data

ChIP-seq data sets (as other high-throughput sequencing data sets) may come in various forms. First, they can appear as a collection of the raw sequencing reads as they come from the sequencing platform, which consists in a list of millions of short DNA sequences, usually from 36 to 50 bp long. For this purpose, the FASTQ format (Cock et al. 2010), that attaches the read quality to the sequences, is used almost exclusively. Second, the reads can be presented already aligned to a reference genome. Essentially, an aligner (also called mapper) software associates each of the reads with a genomic coordinate based on sequence similarity. The SAM and BAM formats (Li and Durbin 2009) are dominant, but individual pieces of software may make use of their own format, as is the case of the GEM mapper (Marco-Sola et al.

2012) used in Chapter 3. Third, mapping data can be aggregated into bins, as in a histogram, thus reporting the number of reads present in each arbitrary genomic window. The advantage of this type of data is that it can be visualized with a genome browser (Kent et al. 2002) using BED or WIG formats. This histogram-like profiles show a higher frequency at or near protein binding sites, consistent with the ChIP principle.

Different proteins generate ChIP-seq signals that are characteristic of how they bind on the genome, but overall there are two main signal types. The archetypical transcription factor profile shows high amplitude peaks in relatively narrow regions of enrichment over the genome. Other proteins that are not transcription factors but also exhibit this binding behavior include CTCF and, to a somewhat lesser extent, H3K4me3. On the other hand, histone modifications tend to bind across broad domains that cover relatively large portions of the genome, forming regions of enrichment of normally lower amplitude than that of peaky signals. It is not unusual for these domains to be on the megabase scale and cover up to 40% of the genome, as is the case of lamin B1, one of the proteins that constitutes the nuclear lamina that also produces this type of profile (Guelen et al. 2008). The differences of binding patterns can be best compared in Fig. 8.

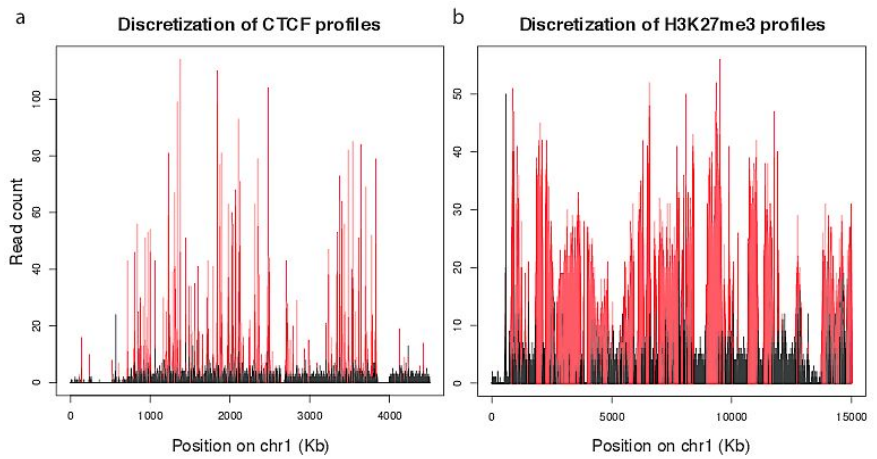

Figure 8. Types of ChIP-seq profiles. The number of reads in each genomic coordinate at the beginning of the human chromosome 1 is shown for two ChIP-seq experiments. The red bars indicate the significantly enriched loci while the black bars represent the remaining, background signal. a) The CTCF profile is a good example of a typical peaky signal. It presents high and narrow peaks and overall a high signal-to-noise ratio. b) The H3K27me3 represents a broad signal common for histone modifications. It also has an overall lower signal-to-noise ratio (note the difference in scale between the two panels).

The read count in each window is a random variable. It is not only affected by whether the protein of interest was bound to the window at the moment of the cross-linking, but also by every other step in the ChIP-seq process. Noise will appear mixed with the signal as a result of many factors (some of which may be unknown) such as nonspecific immunoprecipitation or during DNA amplification or sequencing. Moreover, there are several sources of systematic bias (again, some of them may be unknown), such as the hyper-ChIPable regions that originate from highly expressed loci (Teytelman et al. 2013), as well as amplification-, sequencing- or mapping-related biases. In fact, read counts in ChIP and negative

control samples are usually correlated (Zhang et al. 2008). In order to identify the true signal and recover the protein locations, many software solutions exist and these programs usually receive the generic name of peak callers, peak finders, or signal discretizers. Each implementation uses a different method to find the enriched loci, but they all share a similar objective, to tell the regions that are significantly enriched from the ones that are not, with this significance depending on the assumptions that each model makes.

For instance, MACS (Zhang et al. 2008), one of the most widely used peak callers, uses a predetermined fold enrichment threshold to select a population of candidate regions from which to take a sample and model the genomic distance between the two positions with most reads of each region. This represents the shift in genomic coordinates of reads originated from the two DNA strands, caused by the fact that high-throughput sequencing reads only contain the ends of the DNA fragments obtained in the ChIP-seq experiment. This shift parameter is used to offset the reads on the two strands and then to define a sliding window that scans the genome in search of a significant read count based on a Poisson distribution estimated from the local read distribution in a negative control experiment, or in the same ChIP experiment if negative control is not available (Zhang et al. 2008). This approach represents the state of the art in terms of prediction accuracy, which suggests that the assumptions made by MACS reflect or generate results consistent with a biological truth. It presents, however, a number of limitations. For instance, earlier versions of the software could not reliably detect peaks in broad profiles, the

assumption of the Poisson distribution may not be ideal, and it cannot integrate information from experimental replicates as other software does (Ibrahim et al. 2015). Also, it assumes that the state of enrichment of a genomic window is independent of the state of its neighbors, which may not be necessarily the case, as some peak callers successfully model the chromatin fiber as a Markov process (Spyrou et al. 2009; Qin et al. 2010).

Regardless of the method used for calling the peaks, the predictive accuracy of the calls ultimately depends on the quality of the input data. If the signal-to-noise ratio in the experiment is too low, it will be more difficult to tell the true positives, the peaks will be similar to the background. This could lead to erroneous conclusions as either only a subpopulation of very high confidence peaks will be selected, or a number of spurious, false positive peaks will be included in the call. Also, if the method accounts for evidence found in experimental replicates, and such replicates do not correlate well, it may happen that a fraction of the peaks appear to be reproducible, when in reality they may have been found out of pure coincidence, which would again bias further conclusions. It is thus clear that performing a quality control of some sort constitutes good practice.

In this aspect, the literature is quite sparse, with only one method, the so called Irreproducible Discovery Rate (IDR) (Li et al. 2011), with a widespread use. Briefly, the IDR attempts to assess the degree of reproducibility between experimental replicates, and its results can either be used as a global quality control to accept or discard the experimental data, or as a refinement method to select

the most reproducible peaks. However, it is not exempt of drawbacks: the comparisons between replicates are done in a pairwise manner, which quadratically increases the complexity of the procedure as more replicates become available, it is computationally costly and time consuming and, most importantly, it is not designed nor validated to work on profiles with broad type signal. This last point is especially important, since in the last years a big number of histone modifications with broad domains have been profiled.

CHAPTER 2

Machine learning: how much does it tell about protein folding rates?

Corrales M, Cuscó P, Usmanova DR, Chen H-C, Bogatyreva NS, Filion GJ, et al. Machine Learning: How Much Does It Tell about Protein Folding Rates?. PLoS One. 2015 Nov 25;10(11):e0143166. DOI: 10.1371/journal.pone.0143166

CHAPTER 3

Zerone: a ChIP-seq discretizer for multiple replicates with built-in quality control

Cuscó P, Filion GJ. Zerone: a ChIP-seq discretizer for multiple replicates with built-in quality control. Bioinformatics. 2016 Oct 1;32(19):2896–902. DOI: 10.1093/bioinformatics/btw336

DISCUSSION

Machine learning can be a powerful tool for interpreting and making use of the huge quantities of data currently available. These methodologies have gained popularity precisely because now, more than ever, access to the data allows making findings that were not possible years ago even when using the same algorithms. This increase in popularity has fueled an overconfidence in machine learning that can be seen in studies that claim extraordinary predictive powers resulting from a misuse of such techniques.

By using learning curves and a complex linear model based on amino acid composition we have shown that, in the field of protein folding rates, there is currently not enough data to make predictions as accurate as those reported in the cited studies. This is a straightforward method to diagnose the fit of a learning problem and to know whether the amount of data available is enough for the complexity of the model at hand. As shown in the introduction, learning curves can also be used to determine what is the optimal level of complexity of a model.

The concept of learning curves is not new, but we have found reasons to think that it is not widely used by researchers. It is not only useful to ensure the quality of the study but also to demonstrate the validity of the models in front of the community. We therefore encourage researchers to use and report the results

of using learning curves in fitting supervised machine learning models.

On the other hand, there is a huge amount of ChIP-seq data that can be used in a machine learning setup for the purpose of signal discretization. Many programs have been developed for the task, some of them using techniques that can be considered as machine learning, but these approaches consistently had a number of limitations: it was impossible to integrate information from experimental replicates without using external tools such as IDR. In the case of IDR, it was not developed nor validated to work with broad type signals, which rendered the approach useless for use on the accumulating profiles of histone modifications. Most importantly, IDR was also the only method to validate or reject whole experiments based on replicate reproducibility. Lastly, the computational cost of performing an IDR analysis, especially when more than two replicates are involved, is impractical for high throughput settings.

To address these issues we developed Zerone, a tool that integrates information from experimental replicates and makes a decision by maximum likelihood, weighting the evidence for or against the protein being enriched in every particular genomic window. The most novel feature is its quality control, in which the result of the discretization is evaluated with respect to a curated data set and a decision is made regarding the overall quality of the experiment and the discretization process itself. It is a fast method, well suited for high throughput pipelines in which the researchers must analyze possibly hundreds of experiments. When compared to state-of-the-art discretizers, Zerone shows a competitive

performance. This is partly due to assuming the Markov property in genomic data and, importantly, to the use of the zero-inflated negative multinomial distribution to model read count observations. This allows Zerone to perform well in different types of ChIP-seq profiles and adapt the model to the data set at hand.

Nonetheless, Zerone was developed with a very specific goal in mind, to tell enriched from background genomic windows. This is in contrast to all other discretizers, which focus on finding the exact start and end (and sometimes center) coordinates of each of the peaks. This latter approach poses a challenge when using the discretized data in downstream analyses, like for instance when segmenting the genome into chromatin states based on ChIP-seq signal: it is difficult to properly weigh the number of peaks and their intensity at each genomic location without making risky assumptions about how to adjust the discretizer for different types of signal. With Zerone, even unlikely calls are the most likely based on the data (and if there are too many of these windows the whole discretization may be rejected), which ensures that all profiles are processed in the same way regardless of their signal type or other factors. This was a design decision and it is up to the analyst to decide whether this window-based approach fits her needs.

CONCLUSIONS

The main contributions of this thesis can be summarized as follows:

- We found that exaggeration of model predictive power caused by the misuse of machine learning techniques is an issue in the field of protein folding rates, and demonstrate that model predictive power is currently limited by the availability of data.

- We proposed the use of learning curves to effectively avoid overfitting issues in predicting protein folding rates.

- We developed a machine learning-based ChIP-seq discretizer that integrates replicate information, performs quality control, and overall produces better discretizations than state-of-the-art discretizers.

REFERENCES

Akaike H. A New Look at the Statistical Model Identification. Springer Series in Statistics. 1974;215–22.

Baum LE, Petrie T. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. Annals of Mathematical Statistics. 1966;37(6):1554–63.

BNL Newsroom. First Analysis of Tumor-Suppressor Interactions with Whole Genome in Normal Human Cells Reveals Key Differences with Cancer Cells. Available at: https://www.bnl.gov/newsroom/news.php?a=11351

Brdlik CM, Wei N, Snyder M. Chromatin Immunoprecipitation and Multiplex Sequencing (ChIP-Seq) to Identify Global Transcription Factor Binding Sites in the Nematode Caenorhabditis Elegans. Methods in Enzymology. 2014;539:89–111.

Cock PJ, Fields CJ, Goto N, Heuer ML, Rice PM. The Sanger FASTQ File Format for Sequences with Quality Scores, and the Solexa/Illumina FASTQ Variants. Nucleic Acids Research. 2010;38(6):1767–71.

Dahm R. Friedrich Miescher and the Discovery of DNA. Developmental Biology. 2005;278(2):274–88.

Ernst J, Kellis M. Discovery and Characterization of Chromatin States for Systematic Annotation of the Human Genome. Nature Biotechnology. 2010;28(8):817–25.

Filion GJ, van Bemmel JG, Braunschweig U, Talhout W, Kind J, Ward LD, et al. Systematic Protein Location Mapping Reveals Five Principal Chromatin Types in Drosophila Cells. Cell. 2010;143(2):212–24.

Finch JT, Klug A. Solenoidal Model for Superstructure in Chromatin. Proceedings of the National Academy of Sciences. 1976;73(6):1897–901.

Gilmour DS, Lis JT. In Vivo Interactions of RNA Polymerase II with Genes of Drosophila Melanogaster. Molecular and Cellular Biology. 1985;5(8):2009–18.

Guelen L, Pagie L, Brasset E, Meuleman W, Faza MB, Talhout W, Eussen BH, et al. Domain Organization of Human Chromosomes Revealed by Mapping of Nuclear Lamina Interactions. Nature. 2008;453(7197):948–51.

Hughes, G. On the Mean Accuracy of Statistical Pattern Recognizers. IEEE Transactions on Information Theory / Professional Technical Group on Information Theory. 1968;14(1):55–63.

Ibrahim MM, Lacadie SA, Ohler U. JAMM: A Peak Finder for Joint Analysis of NGS Replicates. Bioinformatics. 2015;31(1):48–55.

International Human Genome Sequencing Consortium. Finishing the Euchromatic Sequence of the Human Genome. Nature. 2004;431(7011):931–45.

Jackson V, Vaughn J. Studies on Histone Organization in the Nucleosome Using Formaldehyde as a Reversible Cross-Linking Agent. Cell. 1978;15(3):945–54.

Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D. The Human Genome Browser at UCSC. Genome Research. 2002;12(6):996–1006.

Li H, Durbin R. Fast and Accurate Short Read Alignment with Burrows-Wheeler Transform. Bioinformatics. 2009;25(14):1754–60.

Li Q, Brown JB, Haiyan H, Bickel PJ. Measuring Reproducibility of High-Throughput Experiments. *The Annals of Applied Statistics.* 2011;5(3):1752–79.

Marco-Sola S, Sammeth M, Guigó R, Ribeca P. The GEM Mapper: Fast, Accurate and Versatile Alignment by Filtration. Nature Methods. 2012;9(12):1185–8.

Medrano-Fernández A, Barco A. Nuclear Organization and 3D Chromatin Architecture in Cognition and Neuropsychiatric Disorders. Molecular Brain. 2016;9(1):83.

Qin ZS, Yu J, Shen J, Maher CA, Hu M, Kalyana-Sundaram S, Yu J, Chinnaiyan AM. HPeak: An HMM-Based Algorithm for Defining Read-Enriched Regions in ChIP-Seq Data. BMC Bioinformatics. 2010;11(July):369.

Rabiner LR. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE 77. 1989;(2):257–86.

Ren B, Robert F, Wyrick JJ, Aparicio O, Jennings EG, Simon I, Zeitlinger J, Schreiber J, Hannett N, Kanin E, Volkert TL, Wilson CJ, Bell SP, Young RA. Genome-Wide Location and Function of DNA Binding Proteins. Science. 2000;290(5500):2306–9.

Ricci MA, Manzo C, García-Parajo MF, Lakadamyali M, Cosma MP. Chromatin Fibers Are Formed by Heterogeneous Groups of Nucleosomes In Vivo. Cell. 2015;160(6):1145–58.

Schwarz G. Estimating the Dimension of a Model. Annals of Statistics. 1978;6(2):461–4.

Solomon MJ, Varshavsky A. Formaldehyde-Mediated DNA-Protein Crosslinking: A Probe for in Vivo Chromatin Structures. Proceedings of the National Academy of Sciences of the United States of America. 1985;82(19):6470–4.

Spyrou C, Stark R, Lynch AG, Tavaré S. BayesPeak: Bayesian Analysis of ChIP-Seq Data. *BMC Bioinformatics.* 2009;10(September):299.

van Steensel B, Henikoff S. Identification of in Vivo DNA Targets of Chromatin Proteins Using Tethered Dam Methyltransferase. Nature Biotechnology. 2000;18(4):424–8.

Stryer L. *Biochemistry* (fourth ed.). New York - Basingstoke: W. H. Freeman and Company; 1995.

Teytelman L, Thurtle DM, Rine J, van Oudenaarden A. Highly Expressed Loci Are Vulnerable to Misleading ChIP Localization of Multiple Unrelated Proteins. Proceedings of the National Academy of Sciences of the United States of America. 2013;110(46):18602–7.

Viterbi A. Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. IEEE Transactions on Information Theory / Professional Technical Group on Information Theory. 1967;13(2):260–9.

Waldminghaus T, Kirsten S. ChIP on Chip: Surprising Results Are Often Artifacts. BMC Genomics. 2010;11(1):414.

Zacharias H. Emil Heitz (1892-1965): Chloroplasts, Heterochromatin, and Polytene Chromosomes. Genetics. 1995;141(1):7–14.

Zhang Y, Tao L, Meyer CA, Jérôme E, Johnson DS, Bernstein BE, et al. Model-Based Analysis of ChIP-Seq (MACS). Genome Biology. 2008;9(9):R137.

ANNEX 1

Supporting Information to Machine learning: how much does it tell about protein folding rates?

Corrales M, Cuscó P, Usmanova DR, Chen H-C, Bogatyreva NS, Filion GJ, et al. Machine Learning: How Much Does It Tell about Protein Folding Rates?. Supplementary material. PLoS One. 2015 Nov 25;10(11):e0143166. DOI: 10.1371/journal.pone.0143166

ANNEX 2

Supplementary Information to Zerone: a ChIP-seq discretizer for multiple replicates with built-in quality control

Cuscó P, Filion GJ. Zerone: a ChIP-seq discretizer for multiple replicates with built-in quality control. Supplementary material. Bioinformatics. 2016 Oct 1;32(19):2896–902. DOI: 10.1093/bioinformatics/btw336