Charles University

Faculty of Science

Study programme: Bioinformatics

Branch of study: BBINF

Lucie Drahoňovská

Peak identification from ChIP-nexus data

Identifikace vazebných míst v datech z ChIP-nexus

Bachelor's thesis

Supervisor: RNDr. Martin Převorovský, Ph.D.

Prague, 2022

Prohlášení:

Prohlašuji, že jsem závěrečnou práci zpracovala samostatně a že jsem uvedla všechny použité informační zdroje a literaturu. Tato práce ani její podstatná část nebyla předložena k získání jiného nebo stejného akademického titulu.

V Praze,

I would like to give my thanks to RNDr. Martin Převorovský, Ph.D. for very understanding and encouraging attitude towards our cooperation, to my parents for very needed alimentary and transport backup, to my dear Evžen Wybitul for putting up with my whining and last but not least Jan Kytka for outstanding emergency technical support.

Abstract

Proteins play a very important role in live organisms. Their roles are for example structural, transport, regulatory or catalytic. What genes will be expressed, what proteins will be made and at what rate can have a strong impact on the function or even health of the organism. Gene expression is significantly regulated by transcription factors, whose activity may cause multiple diseases or disorders (Latchman 1997). Studying those factors and their function is therefore very important. Several methods were developed to this cause, ChIP-chip, ChIP-seq, ChIP-exo and ChIP-nexus. They enable us to study the binding sites of transcription factors and other DNA-binding proteins with various degrees of resolution.
In this thesis I am going to describe the above mentioned methods and peak callers, softwares used for analysis of data obtained by those methods. I will also attempt to do peak calling of ChIP-nexus data of Cbf11 protein and compare the outcomes.

Keywords

Protein-DNA interactions, transcription factors, ChIP-seq, ChIP-exo, ChIP-nexus, peak callers, Cbf11

Abstrakt

Proteiny hrají v živých organismech velice důležitou roli. Zastávají například funkce strukturní, transportní, regulační či katalytické. To, které geny se budou exprimovat a kterým proteinům to dá vzniknout a v jakém množství může mít zásadní vliv na funkci, či dokonce zdraví celého organismu. Regulaci genové exprese mají mimo jiné na starosti transkripční faktory, které tedy svou aktivitou způsobit řadu nemocí, či poruch (Latchman 1997). Jejich studium je tedy velice důležité. K tomuto účelu bylo vyvinuto nekolik technologií, jako je ChIP-chip, ChIP-seq, ChIP-exo a ChIP-nexus, které nám umožňují s větší či menší přesností zkoumat vazebná místa transkripčních faktorů a dalších DNA vazebných proteinů.
V této práci se budu věnovat výše zmíněným technologiím a peak callerům, tedy softwarům využívaným k analýze dat získaných těmito metodami. Také se na nich pokusím provést peak calling ChIP-nexus dat proteinu Cbf11 a porovnat jejich výsledky.

Klíčová slova

Interakce protein-DNA, transkripční faktory, ChIP-seq, ChIP-exo, ChIP-nexus, peak callery, Cbf11

Abbreviations

TF	**transcription factor**
PC	**peak caller**
ChIP	**chromatin immunoprecipitation**
UMI	**unique molecular identifier**
FDR	**false discovery rate**
cut	**cell untimely torn**
DBM	**DNA binding mutant**

Table of contents

Introduction 12

1 Methods 13

- 1.1 ChIP-chip 13
- 1.2 ChIP-seq 15
- 1.3 ChIP-exo 17
- 1.4 ChIP-nexus 19

2 analysis 20

- 1.1 ChIP-chip 21
- 1.2 Chip-seq 21
  - 1.2.1 MACS 21
- 1.3 ChIP-exo 22
  - 1.3.1 Peakzilla 22
  - 1.3.2 MACE 23
- 1.4 ChIP-nexus 23
  - 1.4.1 PeakXus 24
  - 1.4.2 Q-nexus 24

3. Practical part 25

- 3.1 Overview 25
- 3.2 Data 25
- 3.3 MACE 26
- 3.4 PeakXus 26
- 3.5 Peakzilla 26
- 3.6 MACS 26

3.7 Q-nexus 27

4. Discussion 29

References 30

Introduction

Proteins play important roles in all organisms. They can have for example structural, regulatory, catalytic or transportational function. What genes are being expressed, therefore what proteins are being produced and at what rate is crucial for the right function of any organism and even the slightest change in expression may have far reaching consequences. That is why transcription factors (TF) are so important. Transcription factors are proteins which can affect gene expression either positively or negatively. This can happen in multiple ways, but mostly it would be by protein-DNA interaction with the promoter of the regulated gene. Studying the binding sites of TF might tell quite a lot about the TFs and the function of target genes themselves. We might observe similar patterns in the binding sites, which might lead to similar patterns of transcription and regulation (Latchman 1997). Or it can simply tell us what gene the transcription factor interacts with. The best way to study those interactions is *in vivo*, so we can directly see where the transcription factors bind and also under which condition it happens.
In the first part of this thesis I am going to provide a summary of methods that have been used to obtain data for the whole-genome studying of protein-DNA interactions, in the second part I am going to describe the function of several peak callers (PCs), which are used to analyze data obtained from the aforementioned methods, and finally the last part will be dedicated to comparison of results from running the described peak callers on the data from ChIP-nexus experiments for the Cbf11 TF.

1 Methods

1.1 ChIP-chip

The first genome-wide technique for localizing binding sites of transcription factors and other DNA-binding proteins, which has been used since around 2000, is ChIP-chip (chromatin immunoprecipitation analyzed on DNA microarray chip) (Figure 1.). It enables us to study the protein interactions *in vivo* by cross-linking the proteins to their binding sites with formaldehyde, so the bonds are stronger and able to withstand the following processes. Then the DNA is fragmented, usually by sonication, so the resolution of further analysis is enhanced. The next step is the above mentioned immunoprecipitation. For this we need antibodies specific to the target protein. It is used for isolation of fragments which have our protein of interest crosslinked to them. These complexes are then pulled down using polysaccharide or magnetic beads and the rest is washed out. Now, when we have the fragments filtered out, there is no longer a need for the cross-linked proteins, so the cross-linking is reversed and the proteins are removed. At the same time we prepare a control DNA sample from the same chromatin extract, but without the immunoprecipitation process. If needed, samples are amplified by PCR. Finally, samples are labeled, e.g. with fluorescent dyes. Sample DNA is then hybridized onto an oligonucleotide chip array and scanned. The signals obtained from the images can be normalized and filtered, for example using the median filtering (Zheng et al. 2007).

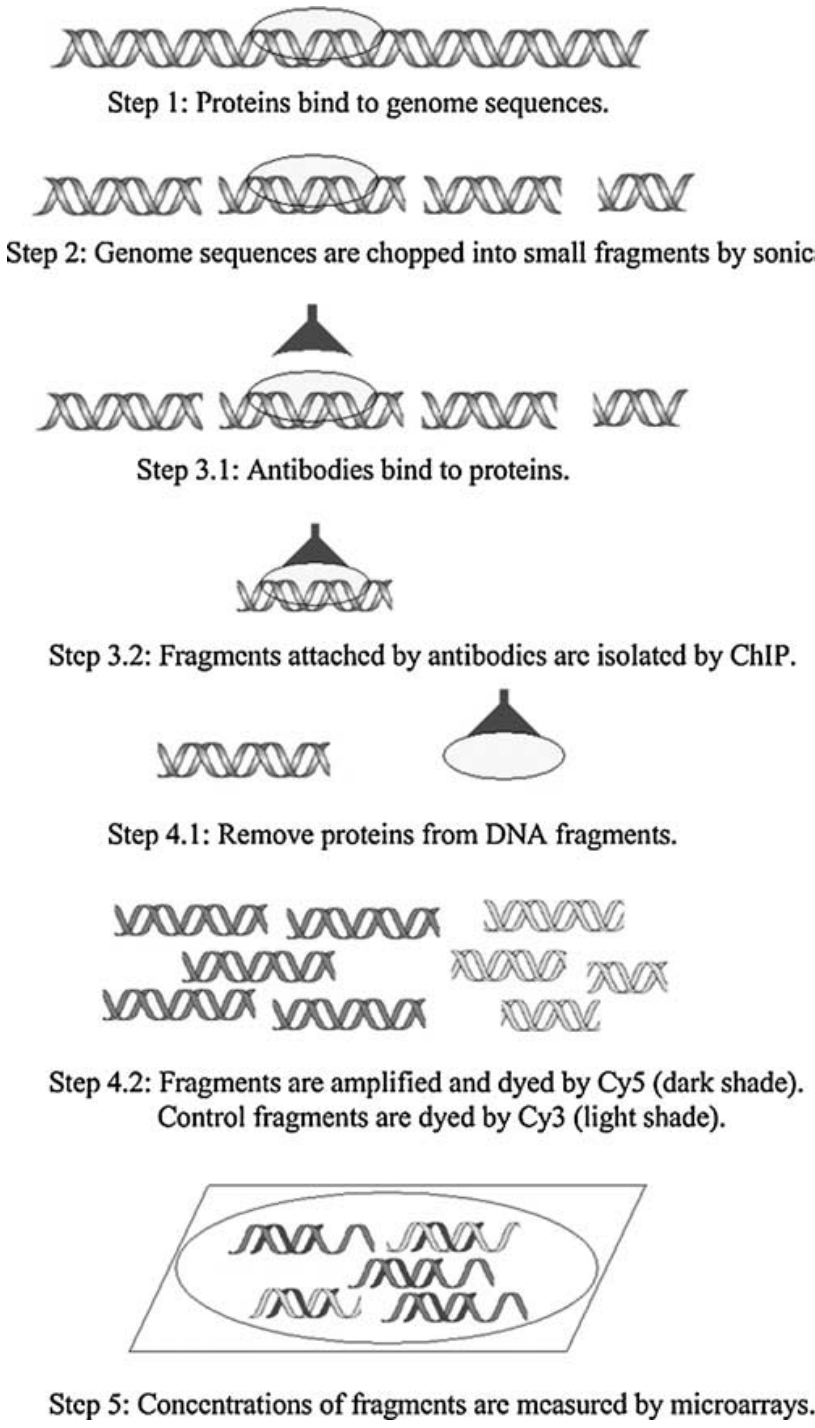

Figure 1. ChIP-chip experiment workflow. Adapted from (Zheng et al. 2007)

ChIP-chip is the first and most basic method for whole-genome studying protein-DNA interactions, so it still has room for improvement. One of its main disadvantages is that hybridization of fragments to the microarray generates noise that is caused by hybridization of imperfectly matched sequences.The genome coverage is limited by the probe library that we use for

creating the microarray, the shorter the spacing between probes, the better the base pair resolution is. But with decreasing spacing and increasing number of probes, the price of the microarray naturally rises too (Zheng et al. 2007). There is also a limited selection of organisms we can study, because we need to have its genome sequenced first so we could design microarray probes for that species. Furthermore the intensity of the signal captured by scanning is not linear, which adds another complication to the analysis (Park 2009; Ho et al. 2011).

1.2 ChIP-seq

The successor of ChIP-chip is a method called ChIP-sequencing (ChIP-seq), which is a union of immunoprecipitation and next-generation sequencing (Figure 2.). The immunoprecipitation part of this procedure is the same as in ChIP-chip, but instead of hybridizing to a microarray, we use next-generation sequencing like Illumina to determine the sequences of once protein-bound fragments of DNA. Also, before the actual sequencing, the immunoprecipitated DNA fragments undergo filtration by size, which should optimally be 150-300 bp (Park 2009).

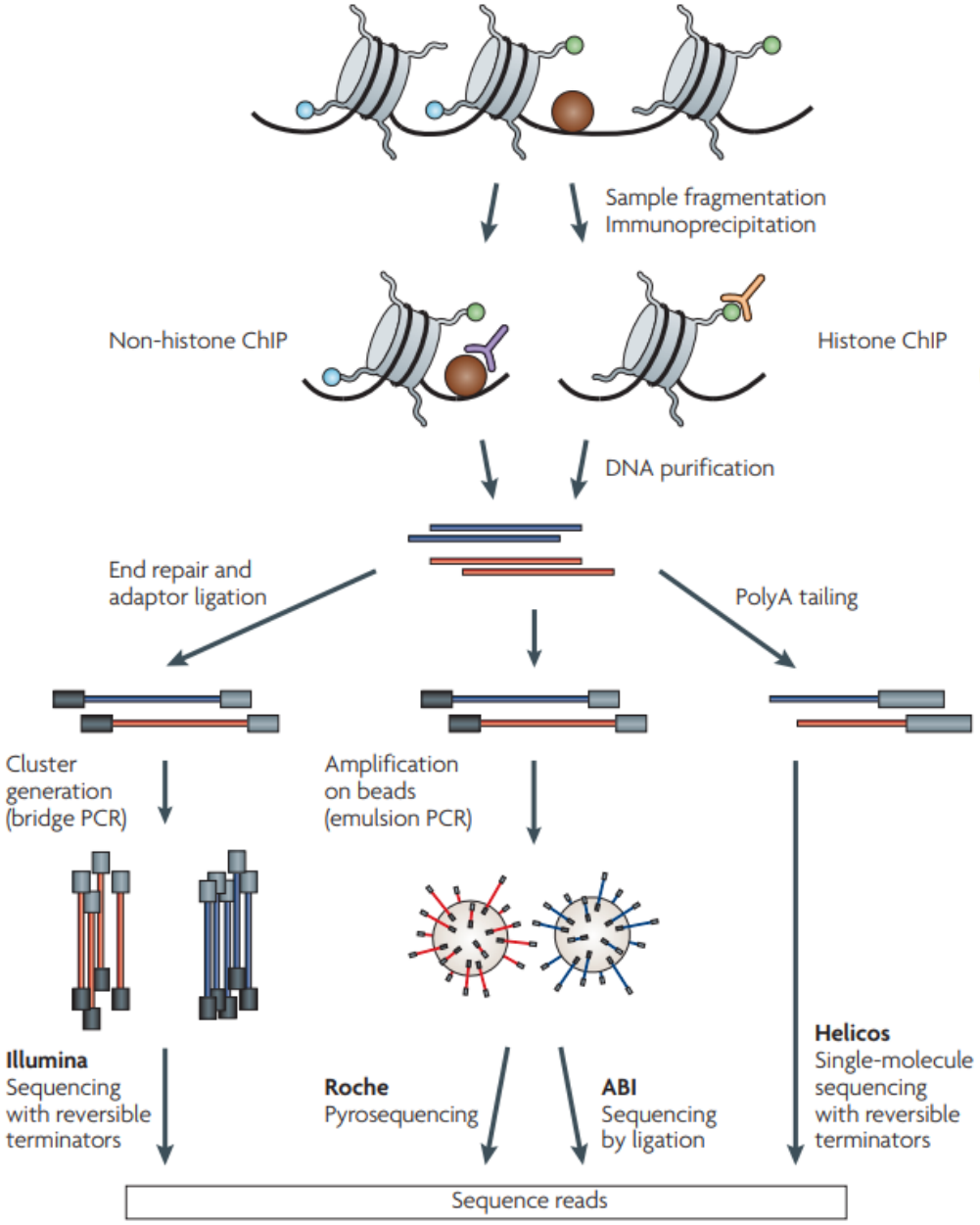

Figure 2. ChIP-seq experiment workflow. Adapted from (Park 2009).

This method has some advantages compared to ChIP-chip. Thanks to the next-generation sequencing, ChIP-seq has higher base pair resolution and enables the use of multiplexing, which requires ligation of a barcode specific for each dataset. The barcodes are then sequenced, allowing for simple attribution of reads to individual samples, and for increased cost-efficiency of the experiments. There is a need for less starting material and it is possible to sequence samples from any organism, no matter the genome size (Park 2009).

On the other hand this method suffers from some errors and biases too. When sequencing, by the end of reads errors occur more frequently. There also might be differences in the frequency of selection of fragments depending on the nucleotide content. GC-poor fragments might be favored in library selection as well as in amplification (Park 2009). And there is one more source of noise. DNA bound nonspecifically to beads might get into the final tag data, and might happen to have stronger signal in some loci than the true, low affinity sites and therefore produce false positives and cause false negatives. (Rhee and Pugh 2011).

1.3 ChIP-exo

Another chromatin immunoprecipitation assay is ChIP-exo. It is quite similar to ChIP-seq however it has some improvements. One of those is the employment of the lambda exonuclease (Figure 3.). During immunoprecipitation of fragmented chromatin, while the DNA is still crosslinked with proteins, an exonuclease is added to the mixture, where it digests DNA from the 5' end. That means that it digests unbound DNA completely, and the cross linked strands up to the region protected by any DNA-binding proteins, where it stops and disconnects. Another difference lies in the adaptors. Since the ligation of adaptors (P2) takes place before the exonuclease treatment, one of the adaptors gets degraded. After the reversion of crosslinks and protease degradation of proteins the P2 adaptor serves as a primer binding site for copying the now single-strand DNA. Then another adaptor is ligated (P1) to the end previously digested by exonuclease and the whole library is sequenced (Rhee and Pugh 2011; Rhee and Pugh 2012).

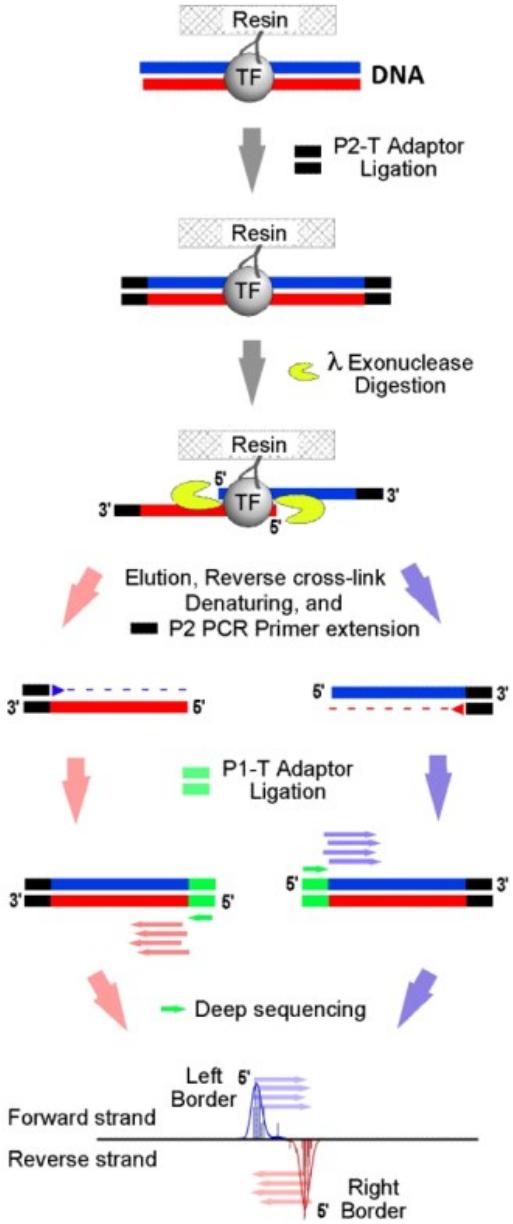

Figure 3. ChIP-exo experiment workflow. Adapted from (Rhee and Pugh 2012)

Thanks to the exonuclease treatment, we get to see the binding site boundaries almost precisely, therefore the resolution of this method is near the single-nucleotide level (Rhee and Pugh 2011). Noise is significantly reduced, because the majority of the unbound DNA fragments, which are the main source of noise in the data, have been removed by the exonuclease. Thanks to that we are able to detect more binding locations, especially the low-occupancy ones, which were hidden in the nonspecific background until now (Rhee and Pugh 2012). This performance however comes with a price, which in this case is the financial costs and computational complexity, which implies long runtime. ChIP-exo also doesn't perform well with low amounts of starting DNA. In that case over-amplification during PCR tends to be a problem. And furthermore this method is not

commercially available, so it can require more laboratory work since we have to prepare all the sequencing libraries ourselves (Kivioja et al. 2011).

1.4 ChIP-nexus

ChIP-nexus is the name of a ChIP-exo variant protocol, which stands for ChIP experiments with nucleotide resolution through exonuclease, unique barcode and single ligation (He et al. 2015). As the title suggests, its main idea is fairly the same as the one of ChIP-exo, just with a few changes. The first difference is that ChIP-nexus uses unique molecular identifiers (UMI) to discover and remove sequences which are over-amplified (Figure 4.). The UMI is a sequence consisting of five random and four fixed bases, which is attached to the beginning of every precipitated DNA fragment, producing a unique sequence combination. Another nexus's trait is a different mechanism of adapters. After immunoprecipitation, an adapter, in fact consisting of the UMI and two adapters separated by a BamHI site, is ligated to both ends of the DNA fragments. UMI are then copied to the complementary strands. Then comes exonuclease digestion, which degrades, among other things, both adaptors on 5' ends. The single-stranded fragments are self-circularized, the BamHI site between the two "subadapters" is digested, thereby linearizing the fragments again, and the finished library fragments are then amplified by PCR (He et al. 2015).

As the ChIP-exo process was quite time-consuming and often yielded low signal, ChIP-nexus aims for better efficiency. The novel adapter approach is a step in the right direction, as one ligation is always more efficient than two ligations. The usage of the UMI barcodes is also a very convenient feature since uncorrected PCR bias may negatively affect the studied data. ChIP-nexus is useful in many biological situations, but it is exceptionally apposite for the study of single nucleotide polymorphisms, which can modify the transcription factor binding. One problem which may occur is that there may not be the same number of reads from both strands. This may be caused by unbalanced efficiency of the lambda exonuclease, ligation efficiency or single-stranded protein-DNA interactions. This imbalance may cause problems, namely in peak calling, since some peak callers depend on the existence of peak-pairs of opposite strand polarity (Welch et al. 2017).

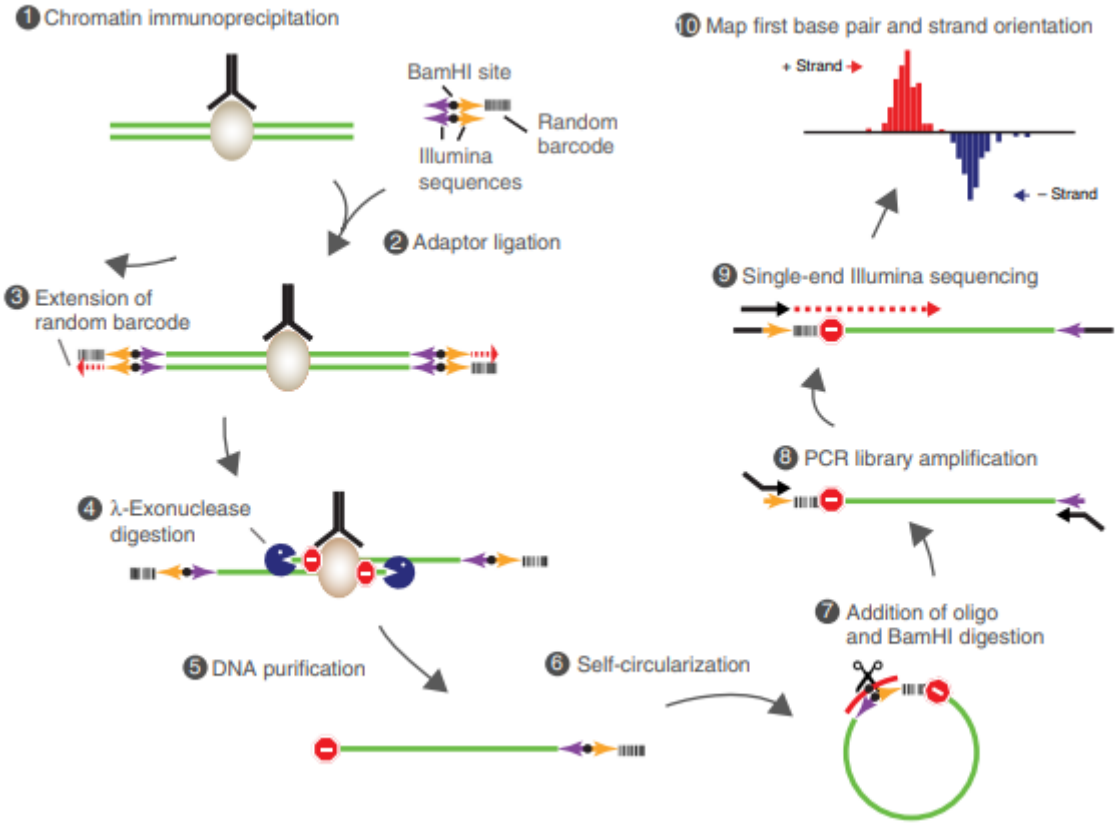

Figure 4. ChIP-nexus experiment workflow. Adapted from (He et al. 2015)

2 Analysis

With the exception of ChIP-chip, the data obtained from the wet-lab procedures usually come in the form of nucleotide sequences of the immunoprecipitated DNA fragments, and need to be further processed. This processing is done by software tools called peak callers. They may use different approaches to the analysis, but their goal is the same. Statistically analyze the data, propose putative TF binding sites by looking for peaks in sequencing read coverage, and test these peaks for significance. Each of the approaches have pros and cons. Some peak callers focus on exploration and endeavor to find all the binding sites even at the cost of high false positive rate. Others may be more conservative and provide just the statistically most significant peaks. Another difference may be the specialization on a specific type of peak. The enriched regions (peaks) come in three different shapes: narrow, broad and mixed, each one typical for a different type of DNA-binding protein (Jeon et al. 2020). Being able to confidently recognize all of those is the best solution, but it is hard to achieve. There are other functionalities such as double peak recognition or implementation of replicates to better detect the noise. Each peak caller is also optimized to a certain degree to the method the analyzed data were generated by. One of the differences is for example the manner of dealing with duplicates. When processing the data from ChIP-seq, two reads that are aligned to the exact same place are most probably the outcomes of PCR

overamplification. When two such reads appear in ChIP-exo data, we can presume those are valid data because of the way they were processed and filtered. In this chapter I will have a look at the means of data analysis of each method from the preceding chapter and describe several peak callers that are currently available.

2.1 ChIP-chip

ChIP-chip data are obtained from the scanned microarray, once for immunoprecipitated DNA fragments and once for the input control ones. Those consist of fluorescence intensities of the probes on the microarray, each corresponding to a specific genomic region. After the data are normalized and filtered a probability model needs to be derived and fitted. There is more than one way to do so, for example Zheng et al. in (Zheng et al. 2007) chose probability model Poisson point process, which modeled potential binding sites (probes) as triangle shapes. From those the algorithm picked local maximum probes, which were the probes with the greatest signal in their neighborhood, and fitted the model at that probe. It repeated this step for all the probes in the neighborhood and then chose the best-fitting triangle-shape which was labeled as a potential binding site. The algorithm stops when all the local maxima are found. The newly found peaks need to be tested. The main idea of the testing is, that if the signals produced by a potential binding site were just noise, it would be possible to model them by a stationary process. We choose a test statistic and use it to compute variance. Then we calculate the P value by comparing the observed signals with signals expected based on the normal distribution with variance from our test statistic. We then discard insignificant peaks based on their P value (Zheng et al. 2007).

2.2 ChIP-seq

ChIP-seq data comes usually in the FASTQ format. Those sequences are then aligned to the genome and stored mainly in the BAM format, or in the CRAM format, which is a newer format with better data compression (Nakato and Sakata 2021). Those are the file formats that peak callers are using as input data. I have chosen MACS as a representative of ChIP-seq peak callers since it is the most frequently used one and it also gives good results in comparison tests (Nakato and Sakata 2021).

2.2.1 MACS

MACS is a software developed for analysis of ChIP-seq data in 2008. Data in the form of mapped reads firstly needs to be filtered. MACS removes duplicate reads that may result from DNA overamplification in PCR. For the correct run of the algorithm it is necessary for both of the samples, the ChIP and the control, to have the same size (i.e., the same number of reads). MACS

therefore, unless instructed otherwise, linearly scales down the larger one. When a DNA fragment is selected during immunoprecipitation, it means that a binding site can probably be found there. And since the reads are typically shorter than the original DNA fragment and because each fragment can be sequenced from either side (strand) with the same probability, the reads form a bimodal enrichment pattern with the peak summit between the modes. The peak is then modeled by extending the reads in the 3' direction until they reach the middle. Then MACS slides a window of a length of double the computed fragment size and searches for enriched regions. It chooses 1000 of them and merges the overlapping ones. Then it builds a model using Poisson distribution, but instead of global lambda it uses dynamic parameter lambda_local, which is counted separately for each candidate peak. This ensures that local fluctuations and biases won't affect the results as much. Based on this model the P value for each potential peak is computed and only those that pass the user-defined threshold are pronounced as peaks. In the next step the false discovery rate (FDR) is calculated. MACS runs its peak-searching algorithm on the ChIP and control sample once more, but this time with swapped control and IP datasets. FDR is then calculated as the ratio of the number of control peaks to the number of ChIP peaks (Feng et al. 2012) (Zhang et al. 2008).

2.3 ChIP-exo

The major difference between ChIP-seq peak callers and the ChIP-exo is that the PCs for ChIP-exo should be prepared for reads represented with just the first nucleotide, which corresponds to the exact boundary of the region protected (i.e., bound) by a DNA-binding protein. They should also be able to recognize the different characteristic traits of PCR duplicates.

2.3.1 Peakzilla

Peakzilla is a peak-calling software developed in 2013 that can analyze data from both ChIP-seq and ChIP-exo experiments. It starts with determining the average fragment size. The procedure of determination depends on the type of data we are dealing with. In the case of paired-end data, the fragment size is simply the average distance between the two outermost points of the read pairs. In the case of single-end data, the fragment length can be computed from the shift size of the positive-strand and negative-strand read peaks taken from the 200 most enriched regions the same way as with the paired-end data. The peak size equals twice the length of the average fragment. All this is done on a dataset with removed PCR duplicates. But for the upcoming operations the original one is used. The model of distribution of positive and negative strand reads is either built from two normal distributions or is estimated empirically using the average distribution of reads within the 200 best candidates. As stated before, the peak calling is done on the complete dataset including

potential PCR duplicates. The reason for this is that the model penalizes the duplicates themselves and in the end it is more sensitive than the preceding deduplication. Each candidate position in the genome gets two scores assigned. One is a count of reads on the positive strand in a window of a size of the average fragment downstream from the candidate and the other is a count of reads on the negative strand in the same window as before, but upstream. The counts are then normalized to a library with 1 000 000 reads and with the corresponding read count from control. Peaks are then assigned as local maxima with at least a fragment size spacing. The peak scores are then corrected with multiplicative distribution which evaluates using the chi-square test how the distribution of computed counts correspond to the distribution expected from the model. Finally FDR is calculated in the same manner as in the case of MACS, which means using the swapped ChIP and control datasets (Bardet et al. 2013).

2.3.2 MACE

MACE is a peakcaller for ChIP-exo data from 2014. It first normalizes the input data's sequencing depth. Then it deals with the bias in nucleotide composition, which happens when position in the read influences the frequency of nucleotides there. A weighting function is used for that, which assigns each read a weight based on the length of the read, proportion of reads that have the same heptamer at the beginning as the weighted read and the proportion of the reads with this heptamer in other positions. Even though ChIP-exo had really reduced the amount of noise by the use of exonucleases, MACE goes beyond it and implements Shannon's relative entropy to combine the signal from multiple replicates, which helps to get the amount of noise even lower. The idea behind that is, that the replicates don't have the exact same noise and therefore by combining them, we can detect the noise and filter it out. The peak calling as such consists of two steps. The first one is searching for peak borders using Chebyshev inequality non-parametric method. The author chose this method because, unlike others, it does not need the information about read distribution or upper and lower boundaries. It is also efficient and robust. In the second step, the algorithm is matching those border peaks together. It is in fact a version of a stable matching problem, which can be solved by the Gale-Shapley algorithm, which guarantees that the matching it finds will be stable. A goodness of a pair is determined based on its coverage score and border-pair size. The coverage score is penalized for any distances deviating from the expected value on either side. Generally we want to maximize the score. Finally we need to combine the P values of the two borders, which is in this case done by the Fisher method (Wang et al. 2014).

2.4 ChIP-nexus

Peak callers for ChIP-nexus do not differ from the ChIP-exo ones a lot, but it is still better to use a dedicated PC to reach the full potential of ChIP-nexus data. The difference lies in the use of UMIs, so the distinguishing feature should be the proper filtration of the data.

2.4.1 PeakXus

PeakXus is a peak caller from 2016 designated for analysis of ChIP-nexus data. Its main feature is that it makes as few assumptions about the data distributions as possible. This allows it to discover new binding patterns, unlike other peak callers (such as MACE or Peakzilla) which are limited by fixed size of peaks, an assumption that often doesn't correspond with the reality. The first step is, as usual, deduplication by filtering identical reads with identical UI. The now unique reads are then divided into signal and background and to those on forward and reverse strand. The binding event is recognizable based on several attributes. First, there are a lot of 5' read ends within a small flanking region around the binding site, thanks to the exonuclease digestion, and second, there are signature borders on opposite strands. The reads that point in a direction other than to the center of the candidate peak are just background. We can also see a pattern in the signal distance from the peak summit, which should be around half the length of the candidate peak, contrary to the background distances, where there shouldn't be any pattern visible, because the reads are positioned randomly. To approximate the read distribution, authors chose G-test, which works with frequency distributions of distances between background and signal reads 5' end, respectively, and the candidate peak summit. This produces a better approximation of chi-square distribution than for example Pearson chi-square test. The only downside of this statistic is that in the equation, we go through all possible distances excluding the ones where the particular distance between background reads and peak summit is not represented by any read. Therefore we lose the information from that locus of the genome even though there may have been a significant number of signal reads. However this problem is easily solvable by adding a pseudocount, which is a small constant added to both distributions that prevents them from acquiring a zero value. After that P value is computed from the adjusted chi-square cumulative distribution function. All candidates with P value greater than 0.05 are pronounced as not a binding event and discarded. Finally an FDR is determined by the Benjamini-Hochberg procedure (Hartonen et al. 2016).

2.4.2 Q-nexus

Q-nexus is a software package from 2016 developed specially for ChIP-nexus. It can handle the deduplication using UMI barcodes and even plot the duplication levels. The main algorithm is based on so-called "qfrags", which are genomic intervals between pairs of read 5' end mapping

positions, one from forward and one from reverse strand, and their distribution. The goal of the algorithm is to estimate the average width of the "protected region", which is simply the DNA section where the protein of our interest binds and thus DNA is protected from exonuclease digestion, and of course to call the peaks. When all the qfrags are assigned, the program looks for local maxima, which become the candidate peaks. Those peaks are then tested for significance by counting the 5' ends of reads that map within a certain radius around the putative peak and computing the P value using Poisson distribution. After that some corrections for multiple testing are done using Benjamini-Hochberg procedure. The candidates are then sorted by P value and the first candidates that have P value lower than a user-specified cutoff are labeled as peaks. Finally the reproducibility of the finished peak calling should be evaluated. For this, authors have chosen an IDR procedure that measures consistency between pairs of biological replicates and is based on peak overlaps (Hansen et al. 2016).

3 Practical part

3.1 Overview

The goal of the practical part of this thesis is to test the performance and user experience of several peak callers. We tested it by analyzing the TF Cbf11 data, which were processed according to the ChIP-nexus protocol. We chose one to two PCs for each method: ChIP-seq, ChIP-exo and ChIP-nexus. In the next paragraphs there is a summary of each of the peak callers sorted increasingly by their approximate success.

3.2 Data

Cbf11 is a transcription factor from the CSL (CBF1/RBP-Jk/Suppressor of Hairless/LAG-1) family from fission yeast *Schizosaccharomyces pombe*. The metazoan family members are part of the Notch receptor signaling pathway, which plays a role for example in the vascular system, haematopoietic system or cell differentiation (Pursglove and Mackay 2005). Cbf11 binds to the promoter of *cut6*, a gene encoding an acetyl-CoA/biotin carboxylase, and activates its expression. Deletion of *cbf11* results in so-called cut (cell untimely torn) phenotype characterized by catastrophic mitosis (Převorovský et al. 2016).
The exact variant we will be interested in, is Cbf11 with a DNA binding mutation (DBM). Because of this mutation it is not able to bind directly to the DNA, but we suspect that it still can get attached to the DNA via some protein complex (Oravcová et al. 2013). In this thesis we want to compare ChIP-nexus data for Cbf11 and Cbf11(DBM) to hopefully gain some more insight into

this issue. Also by comparing Chip-nexus with our published ChIP-seq data, we were hoping to remove any false positive peaks from our old list.
Our data consists of three replicates of non mutated Cbf11 library (Cbf11_1, Cbf11_2, Cbf11_3) and three replicates of Cbf11 with a DNA binding mutation (DBM), which causes significant weakening of the ability to bind directly to DNA (Oravcová et al. 2013). The data (in BAM format) had been already fully pre-processed and mapped .

3.3 MACE

Mace is a peak caller from the ChIP-exo category. It has a wrapper written in Python 2 and the rest of the program, which is accessed through the top one, is in C and C++. Currently, Python version 3.8, which was introduced in 2008, is mainly used. Python 3 is not backward compatible with the now obsolete Python 2 (Malloy and Power 2019), which alone could cause a lot of problems. A lot of newer tools that work well with Python 3 don't work with Python 2 at all. They have to be reinstalled and downgraded to an older version, if any old enough even exists. But what became an unresolvable problem for MACE was that at least one of the C libraries was already deprecated. I haven't succeeded to even install this peak caller on my computer.

3.4 PeakXus

This is one of two peak callers adjusted to analysis of ChIP-nexus data in our selection. And made in 2016, it's also one of the newest ones, so it's already written in Python 3. There weren't many problems with installation. It is possible to run the program, but during the computation there are some errors seemingly caused by improper connection between the files of the programme. It even yields an output, but some files are empty, due to the errors and there is a possibility that the others may be incomplete too.

3.5 Peakzilla

Peakzilla, just like MACE, is designated for ChIP-exo. It is a simple program with quite short source code and not many options. Nevertheless, it states that it is compatible with Python 2.5 or higher, which is confusing, because on my computer it didn't do well with either Python 2 nor with Python 3. On top of that there wasn't an option for peak calling without a negative control sample, which according to the paper introducing MACE (Bardet et al. 2013) should be an option.

3.6 MACS

MACS is a ChIP-seq peak caller and it is the oldest one of this selection. Even Though it is old, it is very well maintained and updated. For this analysis I used MACS3 which worked quite well. The peaks weren't as defined, as they could be with optimal peak caller, but overall it seems like a valid profile. The peak counts were all within the region between 1200 and 2800 which is not much more than we expected.

3.7 Q-nexus

Q-nexus, sometimes called just Q, is the first peak caller which provided complete results. Q is written in C, therefore it has the potential to run visibly faster than those programs written in python (Briggs 2006) . The first run with the default settings was not really satisfying, since there were far more peaks than would be realistically possible and the profile became unreadable (Fig. 6). I used our estimation, that the final peak count would be slightly under one thousand and set *-n* (Maximum number of top peaks to be written to file) from default 100 000 to 1000 and then tweaked -s a little (Number of strand shifts for the determination of the average fragment length. Default: 1000) to focus the peaks even more.
On the figure number 7, there is a cut6 gene zoomed in where we can clearly see with what precision is ChIP-nexus able to locate binding sites.

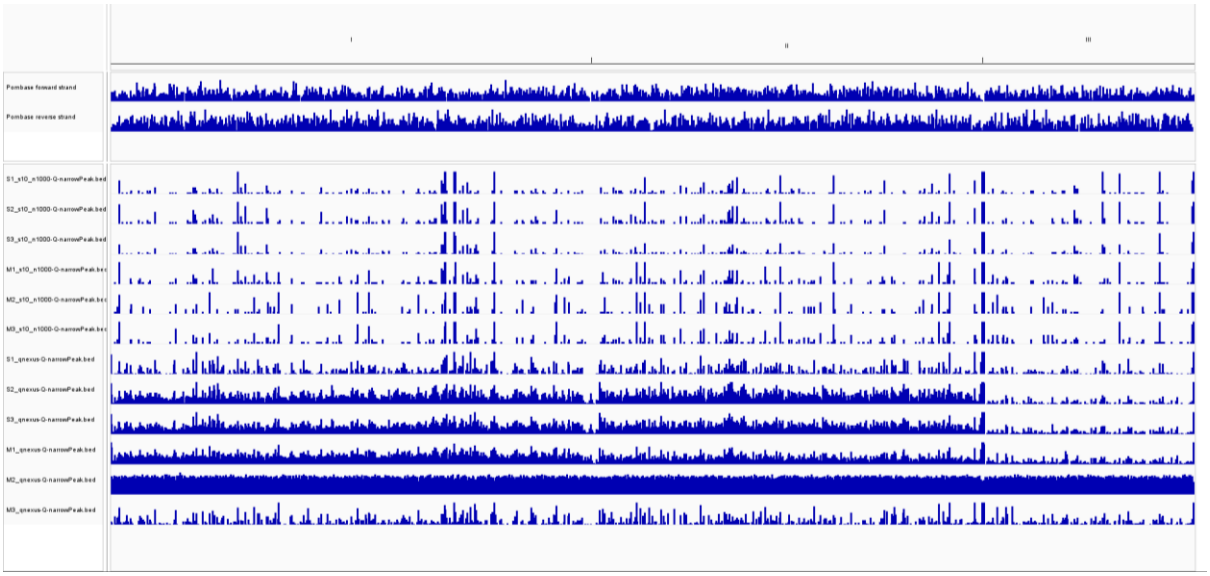

Figure 6.
The first two rows are the two strands of reference genome of *Schizosaccharomyces pombe*, and the rest are the following datasets:
- three default setting peaks of the non mutated Cbf11

- three default setting peaks of the mutated Cbf(DBM)
- three mutated replicates with better settings
- three non mutated with better settings

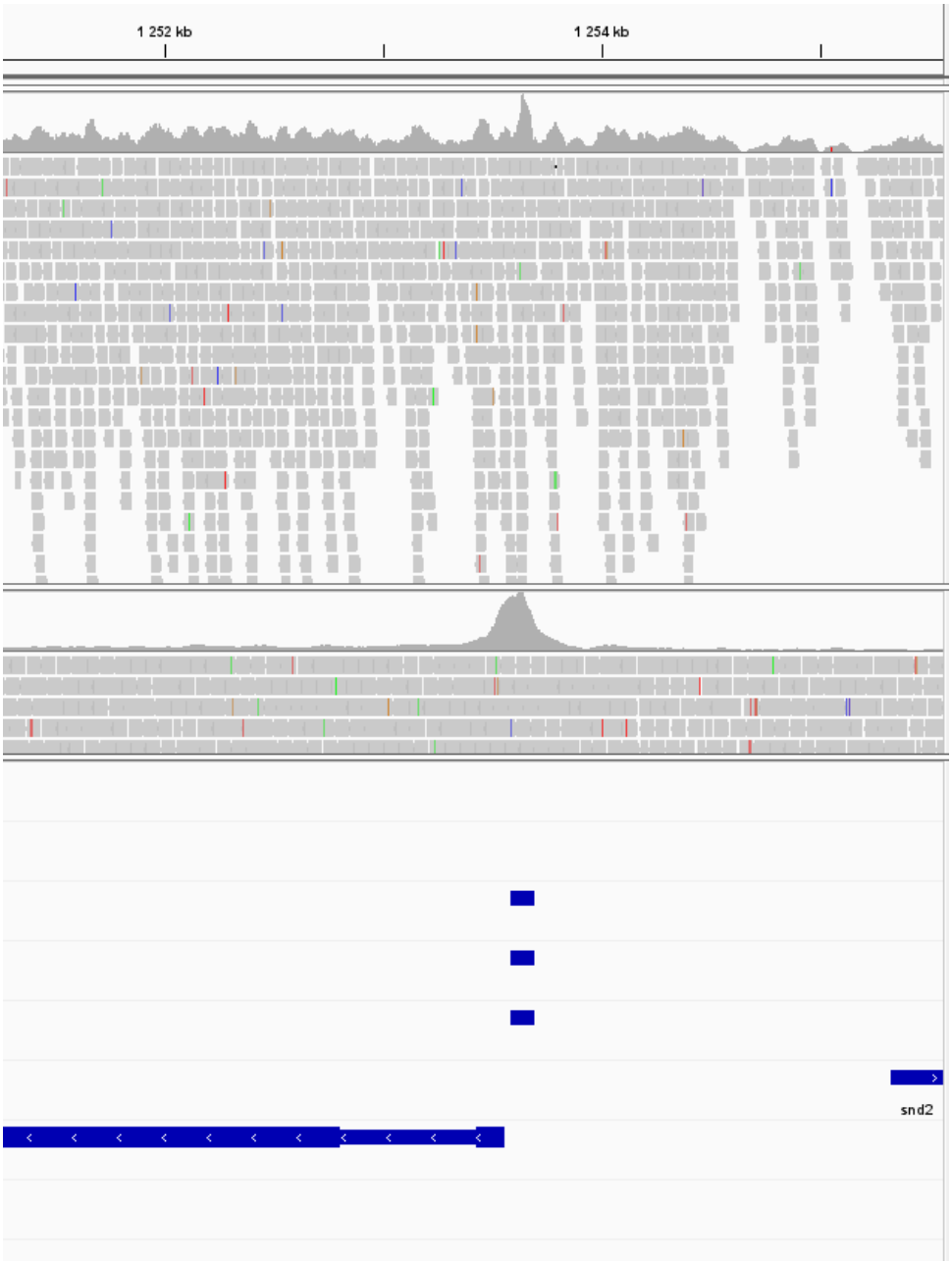

Figure 7.

cut6 gene region of the genome of *Schizosaccharomyces pombe*

- the bottom line is the reference genome
- the next six rows are Cbf11 and Cbf11(DBM) with adjusted settings
- in the area above, there is read coverage of Cbf11 and Cbf11(DBM) respectively

4 Discussion

In this thesis I tried to focus on overall functionality of currently available peak callers and how practical they are for work with ChIP data. The main observation that I have made is that even though their algorithms may still be valuable, the program itself is often neglected. There were peak callers, for example Peakzilla, that were updated for the last time five years ago. Programming languages however are still evolving and such programs may become unusable not in a long time. On the other hand, MACS is in my opinion well kept and despite its age it is in a good shape. What I have also noticed is that the peak callers often lack in documentation. As in installation instructions, or in the code itself. These softwares are in majority free and can be modified and improved by anybody. But it may be quite hard to do so without decent documentation.

We have on the other hand seen peak callers as Q-nexus which yielded results beyond primary expectations. With the use of right parameters it has shown peaks with near-nucleotide resolution. In Figure 7. is shown that the peaks discovered by Q-nexus correspond to the coverage of the BAM file.

But there is still stace for improvement. Ideally, Q-nexus should be able to incorporate the combination of multiple replicates to reduce the noise even more; it should be able to call peaks of different lengths when processing data with various widths of binding places.

References

Bardet, Anaïs F., Jonas Steinmann, Sangeeta Bafna, Juergen A. Knoblich, Julia Zeitlinger, and Alexander Stark. 2013. “Identification of Transcription Factor Binding Sites from ChIP-Seq Data at High Resolution.” Bioinformatics 29 (21): 2705–13.

Briggs, Keith. 2006. “Implementing Exact Real Arithmetic in Python, C++ and C.” Theoretical Computer Science 351 (1): 74–81.

Feng, Jianxing, Tao Liu, Bo Qin, Yong Zhang, and Xiaole Shirley Liu. 2012. “Identifying ChIP-Seq Enrichment Using MACS.” Nature Protocols 7 (9): 1728–40.

Hansen, Peter, Jochen Hecht, Jonas Ibn-Salem, Benjamin S. Menkuec, Sebastian Roskosch, Matthias Truss, and Peter N. Robinson. 2016. “Q-Nexus: A Comprehensive and Efficient Analysis Pipeline Designed for ChIP-Nexus.” BMC Genomics 17 (1): 873.

Hartonen, Tuomo, Biswajyoti Sahu, Kashyap Dave, Teemu Kivioja, and Jussi Taipale. 2016. “PeakXus: Comprehensive Transcription Factor Binding Site Discovery from ChIP-Nexus and ChIP-Exo Experiments.” Bioinformatics 32 (17): i629–38.

He, Qiye, Jeff Johnston, and Julia Zeitlinger. 2015. “ChIP-Nexus Enables Improved Detection of in Vivo Transcription Factor Binding Footprints.” Nature Biotechnology 33 (4): 395–401.

Ho, Joshua W. K., Eric Bishop, Peter V. Karchenko, Nicolas Nègre, Kevin P. White, and Peter J. Park. 2011. “ChIP-Chip versus ChIP-Seq: Lessons for Experimental Design and Data Analysis.” BMC Genomics 12 (February): 134.

Jeon, Hyeongrin, Hyunji Lee, Byunghee Kang, Insoon Jang, and Tae-Young Roh. 2020. “Comparative Analysis of Commonly Used Peak Calling Programs for ChIP-Seq Analysis.” Genomics & Informatics 18 (4): e42.

Kivioja, Teemu, Anna Vähärautio, Kasper Karlsson, Martin Bonke, Martin Enge, Sten Linnarsson, and Jussi Taipale. 2011. “Counting Absolute Numbers of Molecules Using Unique Molecular Identifiers.” Nature Methods 9 (1): 72–74.

Latchman, D. S. 1997. “Transcription Factors: An Overview.” The International Journal of Biochemistry & Cell Biology 29 (12): 1305–12.

Malloy, Brian A., and James F. Power. 2019. “An Empirical Analysis of the Transition from Python 2 to Python 3.” Empirical Software Engineering 24 (2): 751–78.

Nakato, Ryuichiro, and Toyonori Sakata. 2021. “Methods for ChIP-Seq Analysis: A Practical Workflow and Advanced Applications.” Methods 187 (March): 44–53.

Oravcová, Martina, Mikoláš Teska, František Půta, Petr Folk, and Martin Převorovský. 2013. “Fission Yeast CSL Proteins Function as Transcription Factors.” PloS One 8 (3): e59435.

Park, Peter J. 2009. “ChIP-Seq: Advantages and Challenges of a Maturing Technology.” Nature Reviews. Genetics 10 (10): 669–80.

Převorovský, Martin, Martina Oravcová, Róbert Zach, Anna Jordáková, Jürg Bähler, František Půta, and Petr Folk. 2016. “CSL Protein Regulates Transcription of Genes Required to Prevent Catastrophic Mitosis in Fission Yeast.” Cell Cycle 15 (22): 3082–93.

Pursglove, Sharon E., and Joel P. Mackay. 2005. “CSL: A Notch above the Rest.” The International Journal of Biochemistry & Cell Biology 37 (12): 2472–77.

Rhee, Ho Sung, and B. Franklin Pugh. 2011. “Comprehensive Genome-Wide Protein-DNA Interactions Detected at Single-Nucleotide Resolution.” Cell 147 (6): 1408–19.

———. 2012. “ChIP-Exo Method for Identifying Genomic Location of DNA-Binding Proteins with near-Single-Nucleotide Accuracy.” Current Protocols in Molecular Biology / Edited by Frederick M. Ausubel... [et Al.] Chapter 21 (October): Unit 21.24.

Sharma, Vasudha, and Sharmistha Majumdar. 2020. “Comparative Analysis of ChIP-Exo Peak-Callers: Impact of Data Quality, Read Duplication and Binding Subtypes.” BMC Bioinformatics 21 (1): 65.

Wang, Liguo, Junsheng Chen, Chen Wang, Liis Uusküla-Reimand, Kaifu Chen, Alejandra Medina-Rivera, Edwin J. Young, et al. 2014. “MACE: Model Based Analysis of ChIP-Exo.” Nucleic Acids Research 42 (20): e156.

Welch, Rene, Dongjun Chung, Jeffrey Grass, Robert Landick, and Sündüz Keles. 2017. “Data Exploration, Quality Control and Statistical Analysis of ChIP-Exo/nexus Experiments.” Nucleic Acids Research 45 (15): e145.

Zhang, Yong, Tao Liu, Clifford A. Meyer, Jérôme Eeckhoute, David S. Johnson, Bradley E. Bernstein, Chad Nusbaum, et al. 2008. “Model-Based Analysis of ChIP-Seq (MACS).” Genome Biology 9 (9): R137.

Zheng, Ming, Leah O. Barrera, Bing Ren, and Ying Nian Wu. 2007. “ChIP-Chip: Data, Model, and Analysis.” Biometrics 63 (3): 787–96.

Latchman, D. S. 1996. “Transcription-Factor Mutations and Disease.” The New England Journal of Medicine 334 (1): 28–33. https://doi.org/10.1056/NEJM199601043340108