Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

WACS: Improving Peak Calling by Optimally Weighting Controls

Aseel Awdeh[1,2], Marcel Turcotte[1] and Theodore J. Perkins[1,2,3]*

[1]School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, K1N6N5, Canada
[2]Regenerative Medicine Program, Ottawa Hospital Research Institute, Ottawa, ON, K1H8L6, Canada
[3]Department of Biochemistry, Microbiology and Immunology, University of Ottawa, Ottawa, ON, K1H8M5, Canada

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Chromatin immunoprecipitation followed by high throughput sequencing (ChIP-seq), initially introduced more than a decade ago, is widely used by the scientific community to detect protein/DNA binding and histone modifications across the genome. Every experiment is prone to noise and bias, and ChIP-seq experiments are no exception. To alleviate bias, the incorporation of control datasets in ChIP-seq analysis is an essential step. The controls are used to account for the background signal, while the remainder of the ChIP-seq signal captures true binding or histone modification. However, a recurrent issue is different types of bias in different ChIP-seq experiments. Depending on which controls are used, different aspects of ChIP-seq bias are better or worse accounted for, and peak calling can produce different results for the same ChIP-seq experiment. Consequently, generating "smart" controls, which model the non-signal effect for a specific ChIP-seq experiment, could enhance contrast and increase the reliability and reproducibility of the results.
Results: We propose a peak calling algorithm, Weighted Analysis of ChIP-seq (WACS), which is an extension of the well-known peak caller MACS2. There are two main steps in WACS: First, weights are estimated for each control using non-negative least squares regression. The goal is to customize controls to model the noise distribution for each ChIP-seq experiment. This is then followed by peak calling. We demonstrate that WACS significantly outperforms MACS2 and AIControl, another recent algorithm for generating smart controls, in the detection of enriched regions along the genome, in terms of motif enrichment and reproducibility analyses. This ultimately improves our understanding of ChIP-seq controls and their biases, and shows that WACS results in a better approximation of the noise distribution in controls.
Availability: The WACS software is available at https://github.com/aawdeh/MACS/tree/WACS.
Contact: tperkins@ohri.ca, araed104@uottawa.ca

1 Introduction

High throughput sequencing technologies help in uncovering the mechanisms of gene regulation and cell adaptation to external and internal environments [1]. One widely used technology is chromatin immunoprecipitation followed by next generation sequencing (ChIP-seq). It allows the genome-wide investigation of the structural and functional elements encoded in a genomic sequence, such as transcriptional regulatory elements. The main goal of a ChIP-seq experiment is the detection of protein-DNA binding sites and histone modifications genome-wide in various cell lines and tissues. Many peak calling methods have been proposed for the identification of regions of enrichment (putative binding sites) in ChIP-seq data [2, 3, 4, 5, 6].

Every experiment is prone to noise and bias, and ChIP-seq experiments are no exception. While some read pileups correspond to regions of true

enrichment, others may be a result of the distortion of the ChIP-seq signal. Biased or noisy datasets (with a high number of false negative or false positive peaks) negatively impact downstream biological and computational analyses [7]. Thus, accounting for both noise and bias is important. Existing peak callers generally account for noise by assessing statistical significance under some statistical model. Bias is a more complicated subject and is usually addressed explicitly only via some control data to which the ChIP-seq is compared. We return to the issue of controls shortly.

There are many sources of bias in a ChIP-seq experiment. In the experimental design, for example, the quality of the experiment is predetermined by antibody and immunoprecipitation specificity. Low sensitivity, resulting from poor affinity to the target protein of interest, or low specificity, from cross reactivity with other unrelated proteins, degrades the quality of a ChIP-seq experiment [8]. The fragmentation step may also introduce bias [9]. Prior to immunoprecipitation, the DNA-protein complexes undergo fragmentation. However, due to the non-uniform nature of the chromatin structure (DNA), some regions are more densely packed (heterochromatin) than others and are thus more resistant to fragmentation. Less densely packed regions (euchromatin) will undergo more fragmentation. Another source of bias is mappability, which is the extent to which reads are uniquely mapped to regions along the genome [9, 10]. In an ideal situation, long enough reads are used such that there is higher coverage and uniformity in coverage. However, in practice, read length is short and there are "ambiguous" reads that map to multiple regions. Such reads can either be multiply mapped (creating false positive ChIP-seq signal) or discarded (creating empty, unmappable regions), with either choice creating a different sort of bias. GC content bias [11, 12], introduced by PCR amplification or sequencing, also results in imbalanced coverage of reads along the genome. For example, in PCR amplification, GC rich fragments are targeted more than the GC poor fragments. These variations in coverage can have a significant impact on the results obtained.

Systematic and experimental biases hinder the full potential of ChIP-seq analysis. Thus, the quality of the input samples is important, especially in large scale analysis where low quality datasets have greater effects [7, 13]. Consequently, more than a decade after ChIP-seq was introduced, the ENCODE and modENCODE consortia developed a set of ChIP-seq quality control metrics and guidelines to produce high quality reproducible data [8]. The protocols address all the stages of a ChIP-seq experiment, as bias and noise may be introduced at various stages, such as experimental design, execution, evaluation and storage methods [9].

One essential step for the alleviation of bias is the incorporation of control datasets in ChIP-seq analysis. It assists in the selection of true enrichment binding sites from false positives. Controls, such as input DNA, IgG, DNase-seq and/or ATAC-seq, attempt to minimize the effects of immunoprecipitation, antibody imprecision, PCR-amplification, mappability bias, etc., and thereby increase the reliability of the results. In the input DNA, using the same conditions as the original ChIP-seq experiment, the DNA undergoes cross linkage and fragmentation. However, no antibody nor immunoprecipitation is used [8]. For the IgG control, sometimes referred to as a "mock" ChIP-seq experiment, all the same steps and conditions as the original ChIP-seq experiment are applied. However, a control antibody (not specific to the protein of interest) is adopted to interact with non-relevant genomic positions [8]. DNase-seq and ATAC-seq are used to tackle open chromatin regions. According to ENCODE [8], the input DNA and IgG controls should have a sequencing depth greater than or equal to than the original ChIP-seq experiment. Higher sequencing depth is recommended since input DNA signals represent broader genomic chromatin regions than ChIP-seq [8, 9]. Other crucial factors addressed by the protocols include, but are not limited to, biological/technical replicates and library complexity.

Many existing peak calling algorithms allow testing enrichment compared to a control [6, 14, 15, 16, 17, 18, 19]. Whether biases in controls and

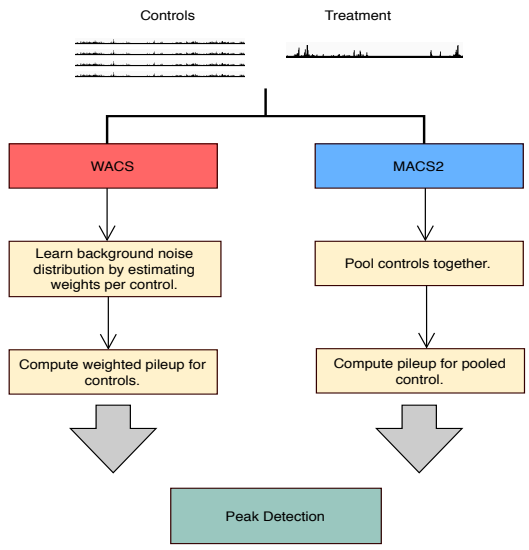

Fig. 1: Flowcharts for WACS and MACS2. Both methods take controls and a treatment as input.

ChIP-seq data are the same is not known, however. None of these methods selects a control or estimates background signals. Depending on which controls are selected and their nature, peak callers can produce different results (i.e., binding site positions) for the same ChIP-seq experiment. The BIDCHIPS [20], CloudControl [21] and AIControl [22] studies have shown that different ChIP-seq datasets can be biased in different ways. They address different biases in different ChIP-seq datasets via the integration of multiple control datasets through regression to improve enrichment analysis. There are some limitations to these studies, however.

For example, BIDCHIPS [20] has the ability to re-prioritize peaks already identified by another peak calling method. However, only five notions of control are accounted for and there are no mechanisms for de novo peak calling based on the combined control [20]. The Hiranuma et al. [21, 22] studies prove the advantage of using more controls to model the background signal. In CloudControl [21], the controls are subsampled in their regression fit proportional to their weights. This then allows the single customized control to be used as input to any peak calling method. However, the downsampling of the combined controls may introduce noise into the control signal.

AIControl [22], a peak calling framework, is an extension of CloudControl [21]. It integrates a group of publicly available control datasets and uses ridge regression to model the background signal. This eliminates the need for the user to input controls. However, some users may want to provide their own controls, and this is not accommodated. Additionally, the number of datasets in ENCODE increases with time, so allowing controls as input in a weighted peak caller is important to represent the newly available datasets and newly explored cell lines.

In this work, we introduce a peak calling algorithm, Weighted Analysis of ChIP-Seq (WACS), which utilizes "smart" controls to model the non-signal effect for a specific ChIP-seq experiment. WACS first estimates the weights for each input control, without requiring the fine-tuning of any parameters. Using the weighted controls, WACS then proceeds to detect regions of enrichment along the genome. WACS is an extension of MACS2.1.1 (Model-based Analysis for ChIP-Seq) [17], the most highly cited open source peak caller. Our development of WACS based on MACS2 allows researchers to use the weighted approach within a peak calling method with which they are familiar, and which has many refined features. Fragment length estimation/detection, read shifting, candidate peak

Algorithm 1 Derive Weights

Input: Control samples (BAM) and ChIP-seq sample (BAM)
Output: Weights per control

1: **procedure** DeriveWeights
2: Index, sort and filter the BAM files.
3: Produce read counts per 200bp windows along genome.
4: Normalize control and ChIP-seq counts by reads per billion.
5: Compute control weights using non-negative linear least-squares regression

Algorithm 2 Peak Detection

Input Control samples (BAM), control weights, ChIP-seq (BAM)
Output Detected peaks

1: **procedure** Peak Detection
▷ **Compute ChIP-seq pileup and associated values**
2: Read treatment sample.
3: Estimate/compute fragment length d based on treatment.
4: *treatmentPileup* ← Compute treatment pileup.
5: $\lambda_{BG} \leftarrow$ *treatReadcount* $\times$ $d \div$ *genomeSize*
6: *LengthScales* ← $[d, 1kb, 10kb]$
▷ **Initialize control pileup to zero at each length scale.**
7: **for** < j = 1 to 3 > **do**
8: *ControlPileup*$[j] \leftarrow 0$
▷ **Read and accumulate each control.**
9: **for** < each control i > **do**
10: Read control i into *FixedWidthTrack*$_i$.
▷ **Loop over length scales.**
11: **for** < j = 1 to 3 > **do**
▷ **Compute scale factor.**
12: $sf \leftarrow d \div LengthScale[j] \times \left(\frac{treatReadcount}{controlReadcount_i}\right)$
▷ **Compute weighted pileup at this scale.**
13: $currentPileup \leftarrow sf \times controlWeight_i \times$ BidirectionalExtendReads(*FixedWidthTrack*$_i$,*LengthScales*$[j]$)
▷ **Add into growing control pileup.**
14: *ControlPileup*$[j] \leftarrow$ *ControlPileup*$[j]+$ *currentPileup*
▷ **Find maximum control pileup.**
15: $\lambda_{local} \leftarrow$ maximum of λ_{BG} and pointwise maximum of *ControlPileup* at three different length scales
▷ **Call peaks.**
16: CallPeaks($treatmentPileup$, λ_{local})

identification, and peak assessment remain the same, while the construction of the control via the weighted combination of datasets is different. To allow for potentially large numbers of controls, we restructure the code invisibly for better memory footprint. We also correct a hashing bug in the pileup-computing code of MACS2, which becomes especially important when we have high read depth and/or many controls. (This bug has subsequently been corrected in the main MACS2 distribution as well.)

We evaluate WACS on 438 ChIP-seq datasets and 147 control datasets from the K562 cell line in the ENCODE database [1]. (The terms ChIP-seq and treatment are used interchangeably throughout the paper.) The results demonstrate the importance of smart bias removal methods and the use of customized control datasets for each ChIP-seq experiment, as the amount of bias varies across different ChIP-seq experiments. In the investigation of downstream genomic analysis, such as motif enrichment and reproducibility, the use of weighted controls in WACS shows a significant improvement in peak detection in comparison with the pooled unweighted controls in MACS2 and weighted controls in AIControl.

2 Methods

2.1 WACS: Weighted Analysis of ChIP-seq

Our approach, WACS, estimates a background distribution by weighting controls, and ultimately identifies regions of enrichment along the genome (Figure 1). Below we describe the five major steps of the WACS algorithm. To implement WACS, we modified a well-known open source algorithm, MACS2. Because there is limited written description of how MACS2 works, we describe some parts of MACS2 to fully describe WACS. The WACS algorithm is summarized into two parts: Derive Weights (Algorithm 1) and Peak Detection (Algorithm 2).

2.1.1 Estimation of weights per control.

The control and treatment samples (in BAM format) are first preprocessed, as seen in Algorithm 1. Using SAMtools [23], we index, sort and optionally filter (remove duplicates from) the BAM files (line 2 in Algorithm 1). We then use BEDtools [24] to convert the BAM files into read counts per 200 base pair (bp) windows along the genome with 50 bp increments (line 3 in Algorithm 1).

Next, WACS normalizes the mapped reads per window for the preprocessed control and treatment samples. This ensures that the control and treatment samples are on the same scale. WACS applies reads per billion normalization to both the control and ChIP-seq samples (line 4 in Algorithm 1). For each sample m and window i:

$$n_{mi} \leftarrow r_{mi} \times 10^9 \div TotalReadCount_m$$

where r_{mi} is the read count in the window, n_{mi} is the normalized read count, and $TotalReadCount_m$ is the total number of reads in sample m. This effectively reproduces the normalization in MACS2, which linearly scales the control sample to the ChIP-seq sample. In what follows, we assume k total controls comprise samples 1 to k, and sample $k+1$ is the ChIP-seq data.

WACS then calculates the weights per input control (line 6 in Algorithm 1). WACS performs non-negative least squares (NNLS) to model the treatment dataset as a function of the controls. The overall objective of the regression is to find the values of the parameters (weights), that minimize the sum of squared differences between predictions and target values, with an additional constraint that allows only positive weights. Given n instances (windows), $y_i = n_{k+1,i}$ target values (one per window), $x_i = (n_{1i}, \ldots, n_{ki})$ feature vectors (one vector per window), a vector Θ of coefficient weights and a constant offset Θ_0, NNLS's objective function is:

$$\min_{\Theta,\Theta_0} \frac{1}{2n} \sum_{i=1}^{n} (y_i - \Theta \cdot x_i - \Theta_0)^2$$

$$\text{subject to } \Theta \geq 0$$

$$\text{and } \Theta_0 \geq 0$$

To solve the NNLS regression we rely on the nnls module from scipy.optimize, part of the scipy [25] package in Python. This produces a weighted control model for the treatment, with weights that indicate the relative importance of each control in modelling the treatment background signal. Zero weights are given to controls not required for modelling the treatment experiment. If there is one control, WACS and MACS2 produce the same output, as by default, the control in WACS gets a weight of exactly 1. The controls can also be weighted by the user, instead of using NNLS to compute the weights of the controls.

2.1.2 Duplicate removal.

Duplicate reads—multiple reads mapped to the same position on the genome—are often due to the overamplification of DNA fragments by

PCR, which leads to the repeated sequencing of a DNA fragment. For WACS and MACS2, duplicate removal is optional. To produce more reliable peak calls, MACS2/WACS remove redundant reads at each genomic locus for both the treatment and control datasets [17]. The default number per genomic locus is determined by the sequencing depth. However, when dealing with multiple controls, MACS2 performs duplicate removal after pooling reads. WACS does the same thing when used in unweighted mode, for the sake of consistency with MACS2. In this case, apparent "duplicates" arising from different sequencing runs may be removed incorrectly, artificially flattening the control read distribution in high density areas. This phenomenon can be particularly prominent when hundreds of controls are being pooled. Thus, we recommend that users who want to perform deduplication do so prior to feeding the mapped read files to MACS2 or WACS.

2.1.3 Fragment length estimation and pileup computation.

For single-end reads, the reads only represent the DNA fragment ends. Thus, to identify the actual regions of enrichment along the genome, the length of the DNA fragment needs to be estimated (line 3 in Algorithm 2). MACS2/WACS [17, 26] begin by sliding a window, with size double the sonication fragment size expected from the ChIP-seq experiment, along the genome to select regions with 5- to 50- fold enrichment relative to a random genome background distribution. MACS2 then randomly selects 1000 regions from these higher quality peaks; these are referred to as model peaks [17, 26]. Surrounding a true enrichment site, the reads show a bimodal distribution for each peak model [17, 26], with forward-strand reads upstream and reverse-strand reads downstream. MACS2/WACS estimate the average length of a DNA fragment *d* by modeling the distance between forward and reverse strand peaks in the bimodal distribution. For treatment pileup computation (line 4 in Algorithm 2), the reads in the ChIP-seq sample are then extended by length *d* in the 5' to 3' direction [17, 26]. Control pileups are computed differently, as we explain shortly.

For paired-end reads, MACS2/WACS do not estimate the fragment length. Rather it is calculated directly from the paired read alignments (line 3 in Algorithm 2). The treatment pileup (line 4 in Algorithm 2) is computed by filling in the intervals between the forward-strand and reverse-strand paired reads.

2.1.4 Computation of weighted pileup for controls.

WACS and MACS2 use a dynamic Poisson distribution per genomic location to model the background distribution [17, 26]. If multiple control samples are given as input, MACS2 pools the controls into one pooled control. WACS, on the other hand, begins by reading the controls into memory one at a time. WACS then uses a weighted mixture of controls to model the background read distribution. Conceptually, for both approaches, pileups are represented as floating point numbers at single base pair resolution along the genome.

Both algorithms use what amounts to a Parzen windows density estimator to smooth the control reads at three different spatial resolutions: *d* (the estimated or calculated ChIP-seq fragment length), 1kbp and 10kbp. For WACS specifically, control densities at each of these length scales are initialized to zero on lines 7 and 8 of Algorithm 2. WACS then loops through each control (line 9). For each control *i*, WACS first reads the BAM/BED file and stores the locations of the reads along the genome in a dictionary $FixedWidthTrack_i$ (line 10 in Algorithm 2). WACS then computes a "scale factor" *sf* depending on the total read counts in the treatment ($treatmentReadcount$) and control ($controlReadcount_i$) sample, and the length scale. Omitting the issue of control weights, the scale factor is such that the Parzen windows estimators at different length scales have the same area under the curve as the treatment pileup.

Next, WACS computes the weighted pileup per control i per scale factor *sf*, as seen in line 13. The control reads are extended in both directions by half the length scale, and then piled up. The pileup is the further weighted by the scale factor and the control weight. Lastly, the weighted pileup is added in the any previous pileups computed at the same length scale (line 14).

Finally, for every basepair in the genome, the total control signal level, λ_{local}, is computed as the maximum of the control densities at each length scale and λ_{BG}, a uniform density over the genome.

2.1.5 Peak detection.

In this step, WACS and MACS2 rescan the genome to find the candidate peaks [17, 26]. (See line 30 in Algorithm 2 - the function takes the treatment and weighted control pileups as input.)

To find peaks (or regions of enrichment), the treatment and control pileups are compared per base pair using a local Poisson test. There are multiple peak calling events across the genome, making this a multiple comparison problem. To limit the false discovery rate, WACS and MACS2 compute q-values (adjusted p-values) for each genomic location using the Benjamini-Hochberg correction [17, 26].

After computing the scores, WACS and MACS2 build the peak regions along the genome. First, regions below the default threshold q-value are discarded. (The user can change the default q-value threshold of 0.05.) The regions above the threshold are then looped through. If the distance between adjacent regions is less than 50bp, the regions are merged into one peak. If the distance between adjacent regions is greater than 50bp, the regions are considered as separate peaks. Next, WACS and MACS2 account for peak length, such that peaks with length less than the estimated fragment length *d* are removed. Finally, a p-value and q-value are associated to each remaining peak based on the summit position—the position within the peak at which the treatment pileup is highest.

3 Results

We evaluated WACS, MACS2.1.1 (https://github.com/taoliu/MACS) and AIControl (https://github.com/hiranumn/AIControl.jl/) on 438 ChIP-seq (treatment) and 147 control samples from the K562 cell line (See Supplementary Table 1 for the accession codes of samples). As seen in Figure 1, MACS2 pools the controls together for each ChIP-seq sample, whereas WACS estimates a weight for each control and computes a unique weighted control pileup for each ChIP-seq sample. AIControl uses a predefined set of publicly available controls [22]. For each ChIP-seq sample, we generated peaks under four conditions: (1) MACS2 with all the 147 controls (All MACS2), (2) MACS2 with the matched ENCODE controls (Matched MACS2), (3) WACS with all the controls (WACS) and (4) AIControl with its predefined controls (AIControl).

3.1 Standardizing peaks for fair comparison of approaches.

Different peak callers can produce peaks in different locations based on the same data, and they can also produce different numbers of peaks. For a fair comparison of the quality of the peaks generated using WACS, MACS2 and AIControl, we adopted the standardization procedure proposed by Hiranuma et al. [22], where the peak width and number of peaks are normalized for each treatment sample. First, the peak width is normalized by binning the peaks in 1000 base pair windows. For example, a peak at chromosome 1 from 14520 to 15420 is counted as two peaks covering bins 14000 to 15000 and 15000 to 16000. Next, the number of peaks under all four conditions for the same dataset is normalized. The top *n* most statistically significant peaks are selected for each of the peak-width normalized datasets, such that *n* is the minimum number of peaks under the four conditions.

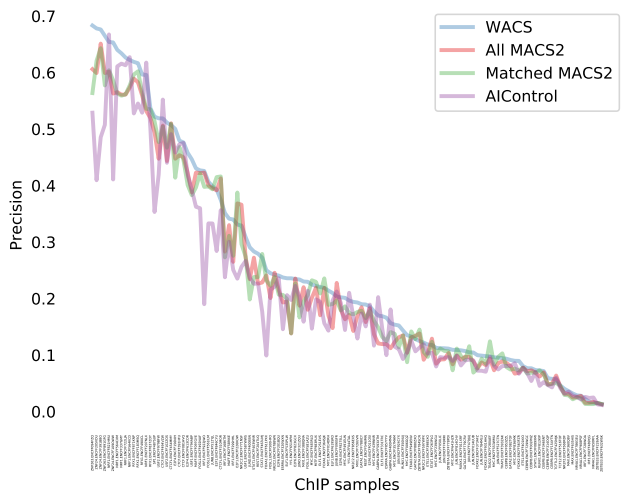

Fig. 2: Precision for the treatment samples, for each of the four peak calling methods. Precision is defined as the fraction of standardized peaks that contain at least one motif occurrence for the transcription factor in question.

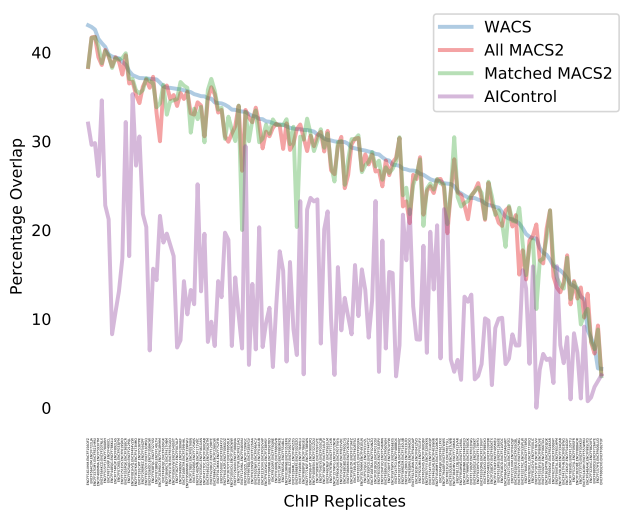

Fig. 3: Percentage overlap in peaks between ENCODE replicates, for each of the four peak calling methods.

3.2 Peaks identified by WACS are more enriched for known sequence motifs.

The purpose of ChIP-seq analysis is the identification of regions of enrichment, such as TF binding sites, along the genome. To evaluate the performance of our method in comparison to MACS2 and AIControl, we performed motif enrichment analysis. Adopting a similar motif analysis method as in [22], we first used JASPAR to obtain position weight matrices (PWMs) for each unique TF [27]. Using PWMs as input, we then used FIMO (Find Individual Motif Occurrences) [28] in the MEME suite [29] to scan the genome and identify motif hits genome wide. We found motifs for 125 treatment samples. Motif hits tend to be enriched in genuine binding sites. In our analysis, peaks with a motif are considered as true positives, while those lacking a motif hit are considered false positives.

Fig 2 displays the precision for each of these ChIP-seq datasets when using WACS (blue line), All MACS2 (red line), matched MACS2 (green line) and AIControl (purple line). WACS outperforms All MACS2, matched MACS2 and AIControl on 113, 107 and 112 of the 125 treatment samples respectively. Using a paired sign test with a two-tailed null hypothesis to compare the precision of WACS with MACS2 and AIControl, we get a p-value of less than 10^{-5}.

The two versions of MACS2 (with all or matched controls) perform similarly to each other. Matched MACS2 slightly outperforms All MACS2, suggesting that the ENCODE selected controls are well-matched to the treatment sample. Across the treatment samples, AIControl has the highest fluctuations in the precision values. The four methods perform similarly on treatment samples when the precision is less than 0.1. This suggests that these treatment samples may need to be more thoroughly investigated. Possibly the sample are of poor quality, or the motifs in JASPAR are not representative of true binding preferences.

Another method for evaluating motif enrichment is the area under the precision-recall curve (AUPRC) [22]. The AURPC is designed to compare algorithms on the same set of instances. Each algorithm, however, generates a different set of peaks for a specific ChIP-seq dataset. Thus, we believe precision is a more appropriate evaluation metric than AUPRC for this comparison. Nevertheless, for the purpose of comparison with AIControl [22] which uses the AUPRC metric, we performed the AUPRC analysis as well. Supplementary Figure 1 shows an example precision-recall curve for the ChIP-seq dataset ENCFF109OWW with TF ZNF24, and Supplementary Figure 2 shows the the AUPRC for each of these ChIP-seq datasets. Using AUPRC, WACS outperforms All MACS2, matched MACS2 and AIControl on 114, 99 and 100 of the 125 treatment samples respectively. These differences are statistically significant by a two-tailed sign test with p-value less than 10^{-5}.

3.3 Peaks identified by WACS are more reproducible.

Ideally, a ChIP-seq peak calling algorithm is able to reproducibly identify true regions of enrichment along the genome with no false positives. Reproducibility is most commonly measured by computing the percentage overlap of peaks between replicates [3, 4]. According to the ENCODE guidelines, each ChIP-seq experiment is associated with at least two ChIP-seq biological replicate samples [1]. Here, we computed the percentage overlap between the standardized peaks for WACS, All MACS2, Matched MACS2 and AIControl across 151 ChIP-seq experiments with two or more replicates. (See Table 2 in Supplementary Data for details.)

Figure 3 shows the percentage overlap for each of the ChIP-seq experiments when using WACS (blue line), All MACS2 (red line), matched MACS2 (green line) and AIControl (purple line). WACS has higher reproducibility than All MACS2, matched MACS2 and AIControl on 107, 108 and 150 of the 151 ChIP-seq experiments respectively. These differences are statistically significant using a paired sign test and at a p-value of less than 10^{-5}.

AIControl has the highest variability, in comparison to MACS2 and WACS, with the percentage overlap fluctuating between 5% and 34%. This shows poor reproducibility between replicates for AIControl. All MACS2 and matched MACS2 have similar performance in terms of percentage overlap. The difference in percentage overlap between WACS and AIControl is noticeably much larger than the difference between WACS and MACS2 across the ChIP-seq replicates.

3.4 Controls used per treatment sample.

Our results (and other results [20, 21, 22]) for motif enrichment and reproducibility analysis suggest that smart controls offer superior background subtraction and peak-calling for ChIP-seq data. However, the standard practice remains to generate controls alongside each ChIP-seq experiment, or

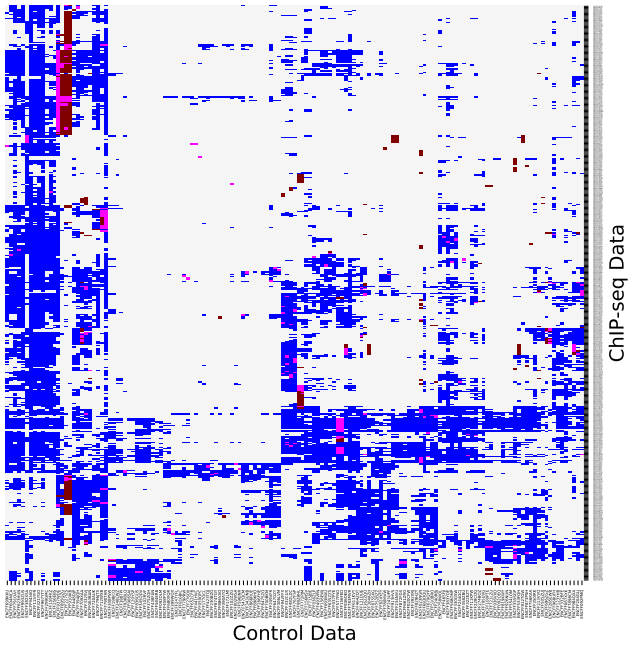

Fig. 4: Comparison of controls used by WACS and ENCODE. The rows and columns correspond to the ChIP-seq and control experiments respectively. For each ChIP-seq dataset, the controls are given a *blue* color if they are used by WACS only, a *maroon* color if they are ENCODE matched controls only, and a *magenta* color if they are used by both ENCODE and WACS.

to match them on the basis of experimental details, such as cell/tissue type, read length and sequencer. If smart controls are to be used, it is unclear how many controls should be considered, and how many will end up in the smart control. It is unclear whether ENCODE matched controls are, in fact, the best choices or even among the controls selected by a smart control procedure.

Here, we aim to increase our understanding of the smart controls used to model the background signal. Figure 4 displays a matrix where the rows and columns represent the ChIP-seq and control datasets respectively. The blue color in the matrix represents the controls selected by WACS to fit each ChIP-seq dataset, the maroon color represents the ENCODE matched controls [1] and the magenta color represents the controls selected by both ENCODE and WACS.

Let us first consider the WACS selected controls per ChIP-seq dataset (blue in Figure 4). Different subsets of the 147 controls are required by WACS for each ChIP-seq dataset, but these form several coherent clusters, where groups of ChIP-seq datasets use relatively the same controls for modeling the background signal. For example, the 10 or so controls most towards the left of the diagram are used in modeling nearly all the ChIP-seq, excepts those towards the bottom. The next 10 controls are widely used, though less so, and are distinct in be used for some of the ChIP-seqs towards the bottom. Conversely, there is a set of ChIP-seq datasets about 1/3 of the way from the bottom they rely on a large number of controls for modeling their background, whereas ChIP-seqs in the upper half rely almost solely on the leftmost controls. Although each ChIP-seq's background is modeled by a unique combination of controls, a clear trend is that many controls are combined—approximately 31 on average, and up to 100 for some samples. Supplementary Figure 3 shows a histogram of the overall number of controls used by the ChIP-seq datasets using WACS.

For the ENCODE matched controls, we observe a range of 1 to 4 ENCODE matched controls per ChIP-seq dataset (maroon color in Figure 4). For 164 of the 438 ChIP-seq datasets (37%), none of the matched ENCODE controls are used to model the background signal in comparison to those used by WACS (rows with no magenta color in Figure 4). For example, 15 controls are used to model the background signal for the ChIP-seq dataset ENCFF025QKV in Figure 4, none of which are the matched ENCODE controls. For the remaining 63% of the ChIP-seq datasets, some of the ENCODE matched controls are also those selected by WACS, as seen in Figure 4 (magenta color). There are 207 ChIP-seq datasets (out of 438) that use all their matched ENCODE controls (in addition to other controls samples), and 32 ChIP-seq datasets that use half of their matched control samples to model the background signal.

4 Discussion

In this paper, we developed a peak calling method, WACS, which allows a mixture of weighted controls as input. The user inputs the controls. These controls can either be weighted by the user, or the weights can be computed by our regression approach. The latter systematically estimates the weights of the input controls to model the background signal for that ChIP-seq experiment. In the special case of equal weights which sum up to 1, the peaks output from WACS and MACS2 are identical. If different weights are allowed, the two algorithms have different outputs. WACS allows only positive weights for better interpretability of results. Negative weights are biologically difficult to interpret; as it does not add to the background signal. WACS proceeds to use this devised background signal to identify regions of enrichment along the genome. WACS is an extension of the most highly cited peak calling algorithm, MACS2 [17]. We conducted a comparison between WACS, MACS2 and AIControl to evaluate our method and the significance of the weighted controls. WACS significantly outperforms both MACS2 and AIControl in motif enrichment analysis and reproducibility analysis.

In addition to provide a method for improved peak-calling, we increased our understanding of ChIP-seq data, controls and their biases. Firstly, we showed that the controls selected by WACS are not necessarily the matched ENCODE controls. Additionally, for most of the ChIP-seq datasets, many more than two controls are selected to model the background signal. This suggests that the ENCODE guidelines may need to be modified to include more than two "matched" controls per ChIP-seq dataset. We showed that a form of intelligent control selection is beneficial for the combination (selection) of controls, as it better models the estimated background signal, where different controls represent different types of biases. As noted by Hiranuma et al. [22], this will also allow researchers to use other controls non-specific to their ChIP-seq experiment to model the noise distribution. This will decrease cost, time and resources required to perform the ChIP-seq experiments. WACS not only provides weights per control for a more efficient background model, it uses an already existing and precise peak calling method in MACS2.

Hiranuma et al. [22] claim that AIControl is better at removing background noise than MACS2. However, our results suggest the contrary (see Figure 3). This may be due to a number of reasons. First, Hiranuma et al. [22] uses a different and nonstandard evaluation method for reproducibility analysis. Whereas we adopted the widely used approach of looking at peak overlaps between biological replicates [3, 4], Hiranuma showed that AIControl had higher irreproducibility than MACS when applied to unrelated datasets. Furthermore, Hiranuma et al. applied MACS2 using only one matched control, while for our analysis, we used either all the ENCODE matched controls for a treatment sample or simply all controls from the same K562 cell line. In either case, the provision of multiple controls may have improved MACS2's performance.

In this manuscript, we described using NNLS to fit a model of ChIP-seq background to control densities, but other formulations are possible. For example, we experimented with an instance-weighted NNLS formulation, to account for differing variances on the regression targets y_i (the ChIP-seq read counts per window). We did not find any improvement in performance. However, results may depend on how one estimates target variances. Relatedly, performing regression on log-transformed read counts may be worth exploring. RNA-seq analysis tools such as DESeq2 [30] use log linear models for read counts and comparisons between conditions. It would also make sense to explore L1-penalized regression formulations, to explore trade offs between the number of controls used to model background and the accuracy of the background model.

Future work will deal with a more thorough analysis of the weighted controls approach on other high throughput sequencing data, such as RNA-seq, and other cell lines, such as GM12878 and HepG2. The weighted approach will be used to study the biases in RNA-seq data across different platforms, labs, cell types, tissues, etc. For example, RNA-seq is used to measure the difference in gene expression between tissues, where a tissue consists of a mixture of cell types. To generate a realistic control tissue, the weighted approach can be used to weight the cell types in the tissue to model the background signal. Also, in this analysis, we focused on sharp peaks, which are more generally found at protein-DNA binding sites. Thus, an analysis of other broader peaks, for example, will be conducted. Ultimately, the overall aim is to further our understanding of the significance of the increasing amount of publicly available data in peak calling analysis to obtain more efficient results.

5 Acknowledgments

This work was supported in part by a Queen Elizabeth II Graduate Scholarship in Science and Technology (QEII-GSST) to AA, and by an NSERC Discovery Grant to TJP.

References

[1]ENCODE Project Consortium et al. *Science*, 306(5696):636–640, 2004.

[2]S. Pepke, B. Wold, and A. Mortazavi. *Nat. Methods*, 6(11s):S22, 2009.

[3]T. D. Laajala, S. Raghav, S. Tuomela, R. Lahesmaa, T. Aittokallio, and L. L. Elo. *BMC genomics*, 10(1):618, 2009.

[4]A. F. Bardet, Q. He, J. Zeitlinger, and A. Stark. *Nat. Protoc.*, 7(1):45, 2012.

[5]E. G. Wilbanks and M. T. Facciotti. *PloS one*, 5(7):e11471, 2010.

[6]R. Thomas, S. Thomas, A. K. Holloway, and K. S. Pollard. *Brief. Bioinform*, 18(3):441–450, 2016.

[7]G. K. Marinov, A. Kundaje, P. J. Park, and B. J. Wold. *G3*, 4(2):209–223, 2014.

[8]S. G. Landt, G. K. Marinov, A. Kundaje, P. Kheradpour, F. Pauli, S. Batzoglou, B. E. Bernstein, P. Bickel, J. B Brown, P. Cayting, et al. *Genome Res.*, 22(9):1813–1831, 2012.

[9]C. A. Meyer and X S. Liu. *Nat. Rev. Genet.*, 15(11):709–721, 2014.

[10]M. Karimzadeh, C. Ernst, A. Kundaje, and M. M. Hoffman. *bioRxiv*, page 095463, 2016.

[11]Y. Benjamini and T. P. Speed. *Nucleic Acids Res*, page gks001, 2012.

[12]M. Teng and R. A. Irizarry. *bioRxiv*, page 090704, 2016.

[13]R. Nakato and K. Shirahige. *Brief. Bioinform*, page bbw023, 2016.

[14]A Mortazavi, BA Williams, K McCue, L Schaeffer, and B Wold. Nat meth 2008; 5 (7): 621-8, 07; pmid: 18516045.

[15]A. P. Fejes, G. Robertson, M. Bilenky, R. Varhol, M. Bainbridge, and S. JM Jones. *Bioinformatics*, 24(15):1729–1730, 2008.

[16]C. Zang, D. E. Schones, C. Zeng, K. Cui, K. Zhao, and W. Peng. *Bioinformatics*, 25(15):1952–1958, 2009.

[17]Y. Zhang, T. Liu, C. A. Meyer, J. Eeckhoute, D. S. Johnson, B. E Bernstein, C. Nusbaum, R. M. Myers, M. Brown, W. Li, et al. *Genome Biol.*, 9(9):R137, 2008.

[18]A. Harmanci, J. Rozowsky, and M. Gerstein. *Genome Biol.*, 15(10):474, 2014.

[19]J. Rozowsky, G. Euskirchen, R. K. Auerbach, Z. D. Zhang, T. Gibson, R. Bjornson, N. Carriero, M. Snyder, and M. B. Gerstein. *Nat. Biotechnol.*, 27(1):66, 2009.

[20]P. Ramachandran, G. A. Palidwor, and T. J. Perkins. *Epigenetics & chromatin*, 8(1):33, 2015.

[21]N. Hiranuma, S. Lundberg, and S. Lee. In *Proceedings of the 7th ACM BCB*, pages 191–199. ACM, 2016.

[22]N. Hiranuma, S. M. Lundberg, and S. Lee. *bioRxiv*, page 278762, 2018.

[23]H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, and R. Durbin. *Bioinformatics*, 25(16):2078–2079, 2009.

[24]A. R. Quinlan and I. M. Hall. *Bioinformatics*, 26(6):841–842, 2010.

[25]E. Jones, T. Oliphant, and P. Peterson. 2014.

[26]J. Feng, T. Liu, B. Qin, Y. Zhang, and X. S. Liu. *Nat. Protoc.*, 7(9):1728, 2012.

[27]A. Khan, O. Fornes, A. Stigliani, M. Gheorghe, J. A. Castro-Mondragon, R. van der Lee, A. Bessy, J. Cheneby, S. Kulkarni, G. Tan, et al. *Nucleic Acids Res*, 46(D1):D260–D266, 2017.

[28]C. E. Grant, T. L. Bailey, and W. S. Noble. *Bioinformatics*, 27(7):1017–1018, 2011.

[29]T. L. Bailey, M. Boden, F. A. Buske, M. Frith, C. E. Grant, L. Clementi, J. Ren, W. W. Li, and W. S. Noble. *Nucleic Acids Res*, 37(suppl_2):W202–W208, 2009.

[30]