BIOINFORMATICS ORIGINAL PAPER

Vol. 30 no. 24 2014, pages 3467–3475
doi:10.1093/bioinformatics/btu722

Genome analysis

Advance Access publication November 3, 2014

Detecting differential peaks in ChIP-seq signals with ODIN

Manuel Allhoff[1,2,3,4], Kristin Seré[2,3], Heike Chauvistré[2,3], Qiong Lin[2,3], Martin Zenke[2,3] and Ivan G. Costa[1,2,3,4,5,*]

[1]IZKF Computational Biology Research Group, RWTH Aachen University Medical School, Germany, [2]Department of Cell Biology, Institute for Biomedical Engineering, RWTH Aachen University Medical School, Germany, [3]Helmholtz Institute for Biomedical Engineering, RWTH Aachen University, Germany, [4]Aachen Institute for Advanced Study in Computational Engineering Science (AICES), RWTH Aachen University, Germany and [5]Center of Informatics, Federal University of Pernambuco, Brazil

Associate Editor: Gunnar Ratsch

ABSTRACT

Motivation: Detection of changes in deoxyribonucleic acid (DNA)–protein interactions from ChIP-seq data is a crucial step in unraveling the regulatory networks behind biological processes. The simplest variation of this problem is the differential peak calling (DPC) problem. Here, one has to find genomic regions with ChIP-seq signal changes between two cellular conditions in the interaction of a protein with DNA. The great majority of peak calling methods can only analyze one ChIP-seq signal at a time and are unable to perform DPC. Recently, a few approaches based on the combination of these peak callers with statistical tests for detecting differential digital expression have been proposed. However, these methods fail to detect detailed changes of protein–DNA interactions.

Results: We propose an One-stage DIffereNtial peak caller (ODIN); an Hidden Markov Model-based approach to detect and analyze differential peaks (DPs) in pairs of ChIP-seq data. ODIN performs genomic signal processing, peak calling and *p-value* calculation in an integrated framework. We also propose an evaluation methodology to compare ODIN with competing methods. The evaluation method is based on the association of DPs with expression changes in the same cellular conditions. Our empirical study based on several ChIP-seq experiments from transcription factors, histone modifications and simulated data shows that ODIN outperforms considered competing methods in most scenarios.

Availability and implementation: http://costalab.org/wp/odin.

Contact: ivan.costa@rwth-aachen.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on April 6, 2014; revised on October 3, 2014; accepted on October 24, 2014

1 INTRODUCTION

Transcriptional regulation is dictated by a set of deoxyribonucleic acid (DNA) interacting proteins such as histones, which define the chromatin structure, and transcription factors (TFs). These proteins are responsible for recruitment of ribonucleic acid (RNA) polymerase and start of transcription (Maston *et al.*, 2006). Detection of the changes in DNA–protein interactions under distinct cellular conditions is a crucial step in unraveling the regulatory networks behind biological processes such as cell differentiation, the activation of signaling pathways and the onset of diseases (Kaikkonen *et al.*, 2013; Martens and Stunnenberg, 2013). Chromatin immunoprecipitation followed by DNA sequencing (ChIP-seq) allows to study the interactions of proteins with DNA regions in a genome wide manner (Dahl and Collas, 2008).

Computational methods identify DNA–protein interaction sites from ChIP-seq (also called peak detection) by detecting regions with a higher number of reads than expected by chance using sophisticated statistical models (Ashoor *et al.*, 2013; Kuan *et al.*, 2011; Rozowsky *et al.*, 2009; Song and Smith, 2011; Spyrou *et al.*, 2009; Zhang *et al.*, 2008;). With a few exceptions, widely used peak calling methods are based on the analysis of individual ChIP-seq signals. Here, we are interested in comparing ChIP-seq data from distinct experiments; we want to detect changes, which we call differential peaks (DPs), in protein–DNA interaction of a single protein in a pair of cellular conditions. We will refer to this signal detection challenge as differential peak calling (DPC). See Figure 1 for examples of putative DPs after the induction of TLR4 signaling (Kaikkonen *et al.*, 2013).

Initially, DPC was performed by peak calling on individual ChIP-seq signals. Peaks detected in only one of the conditions were then defined as cell-specific peaks (Heinz *et al.*, 2010). Such methods are not able to detect cases where peaks were presented (and called) in both cell types, but exhibit a significant increase (decrease) of the DNA–protein signal in one of the cells. In the example of Figure 1, based on peaks from PeakSeq (Rozowsky *et al.*, 2009), only DP1 would be detected as cell-specific. Moreover, most of single signal peak callers (SPCs) provide no functionality to normalize several ChIP-seq experiments and are likely to show bias in experiments with higher number of reads.

A more sophisticated strategy to detect DPs is the combination of peaks from SPCs with statistical tools. These two-stage differential peak callers (DPC) first combine peaks that are called on individual ChIP-seq conditions using SPCs. Next, they count the number of reads for each candidate peak, perform signal normalization and apply statistical tests assuming a differential count model like EdgeR (Robinson *et al.*, 2010) or DESeq (Anders and Huber, 2010). Therefore, they can detect

*To whom correspondence should be addressed.

© The Author 2014. Published by Oxford University Press. All rights reserved. For Permissions, please e-mail: journals.permissions@oup.com

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

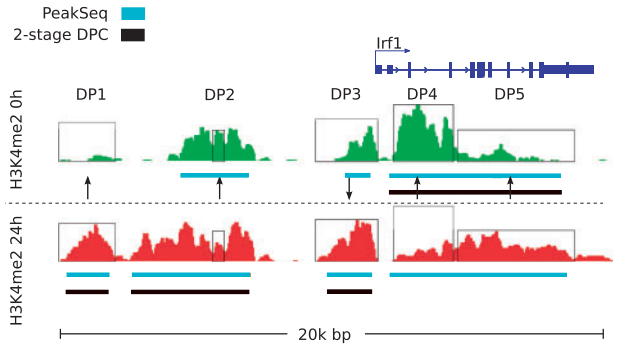

Fig. 1. We show an example of two distinct ChIP-seq signals for the histone modification H3K4me2 before (0 h, upper signal) and 24 hours after (24 h, lower signal) induction of TLR4 signaling of macrophages around the gene Irf1. We indicate with squares examples of regions, which are putative DPs with gain (or loss) of ChIP-seq signal after 24 hours of TLR4 treatment. The height of the squares indicates the size of the highest ChIP-seq signal for a DP. We also display results from the SPC PeakSeq and a two-stage peak caller based on applying DESeq on PeakSeq peaks (bars below the ChIP-seq signals). PeakSeq successfully detects broad peaks describing ChIP-seq signal for each cell. The two-stage peak caller can detect DP1 and DP3, but cannot detect changes within the broad candidate peaks such as DP2 or complex changes in the signal around Irf1 gene body (DP4 and DP5)

candidate peaks where the number of read counts is significantly higher/lower in one of the ChIP-seq conditions. While this approach allows the detection of significant changes in ChIP-seq within candidate peaks, it is highly dependent on the initial peak calling step as well as the strategy used to create the set of candidate peaks. For example, histone modifications associated to active regulatory regions occur in domains spanning several hundreds of base pairs and may have intricate patterns of gain/loss of ChIP-seq signals within the same domain. SPCs tend to call the domains as single peaks and consequently the differential analysis is only able to evaluate the differential counts of the complete called peaks (Fig. 1). A more appropriate framework to analyze DPs is the use of segmentation methods like Hidden Markov Models (HMMs), which are able to analyze pairs of ChIP-seq signals and perform DPC in a single step (Xu *et al.*, 2008).

1.1 Previous approaches

Previous approaches of DPC can be categorized in two classes: one-stage and two-stage DPC. DBChIP (Liang and Keleş, 2012) is an example of a two-stage DPC, as it receives as input the summit information of peaks from SPCs like PeakSeq followed by the application of edgeR (Robinson *et al.*, 2010). DBChIP has as objective only the analysis of TF peaks, and therefore, uses predefined short regions around the peak summits as candidates for DPs. We have previously applied a similar approach using the peak caller MACS (Zhang *et al.*, 2008) and EdgeR accordingly (Lin *et al.*, in preparation). Here, peaks for all ChIP-seq conditions analyzed were combined by merging overlapping regions, which allows the analysis of peaks of any size. Another approach, MAnorm (Shao *et al.*, 2012), also receives as input the regions from SPCs and normalizes the peak counts between two samples with a local robust regression approach followed by a *P* value calculation to define significant DPs.

One-stage DPCs are based on signal segmentation approaches such as an HMM. To our knowledge, the only published method proposed specifically for the DPC problem is ChIPDiff (Xu *et al.*, 2008), which is based on a three state HMM, to detect DPs between two samples. The HMM emission is based on an approximation of a Beta-Binomial distribution, which is fixed after model initialization. The Baum–Welch algorithm is used to estimate transition parameters. Furthermore, the peak caller MACS has been recently extended to MACS2. MACS2 is able to perform DPC of ChIP-seq signal pairs. It works by first performing peak calling in individual signals followed by a fold change analysis within these regions (unpublished, available at https://github.com/taoliu/MACS/). RSEG (Song and Smith, 2011) is specialized in the analysis of single ChIP-seq signals of repressive histones, which are distributed in very large and poorly sequenced genomic domains. It has an option for DPC with a three state model similar to ChIPDiff. RSEG uses Difference Negative Binomial distribution. There are also SPCs based on HMMs, such as HMCan (Ashoor *et al.*, 2013) and BayesPeaks (Spyrou *et al.*, 2009), which are not able to perform DPC.

1.2 Our approach

Here, we propose an One-stage DIffereNtial peak caller (ODIN), an HMM-based approach to detect and analyze DPs in pairs of ChIP-seq data. The HMM, which is based on a three state topology, uses the product of Binomial or mixture of Poisson distributions to model the observed values. Moreover, we constrain several HMM parameters to decrease the number of free parameters and to improve robustness and avoid label switching. Previous to the HMM application, we propose a pipeline with several steps for generation of the ChIP-seq profiles. The pipeline consists of automatic estimation of fragment size, DNA mappability, GC-content normalization, correction of signals with input-DNA and signal normalization. Moreover, we use the density estimates from the HMM for an efficient approach to estimate *P* values for the DPs. ODIN is the first DPC performing signal normalization, DP detection and statistical testing in an integrated way. We propose here a methodology and statistic to evaluate DPs, which is based on associating DPs with changes in expression on the same cellular conditions as the ChIP-seq data. This allows us to perform a comparative analysis with all competing methods (DBChIP, MAnorm, DESeq, MACS2 and ChIPDiff) for ChIP-seq data from TFs and histone modifications. The analysis also covers a simulated data approach tailored for the DPC problem. We use the simulated data to explore the method performance under distinct conditions, such as the complexity of DPs and the number of reads.

2 METHOD

2.1 Profile construction

The first step in ChIP-seq analysis is the construction of a genomic profile. The basic idea is to fragment the genome into bins and count

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

the reads falling in these bins. Let $\mathbf{X}$ be the matrix that represents the genomic signal

$$\mathbf{X} = \{x_{ij}\}^{D \times L},$$

where x_{ij} indicates the number of reads in bin j of signal i, and where D is the number of genomic signals and L the number of bins. Here, we will have $D = 2$, as we are interested in DPs between two ChIP-seq signals. The ith genomic signal is represented by the row vector $x_{i\cdot} = \{x_{i1}, \ldots, x_{iL}\}$ and the genomic signals for bin j is represented by the vector $x_{\cdot j} = \{x_{1j}, \ldots, x_{Dj}\}$. Moreover, ChIP-seq experiments usually have reads from input-DNA for each cell type analyzed. The input-DNA contains all fragmented DNA not immunoprecipitaded in the ChIP experiment and can be used as a control signal. In particular, input-DNA can indicate sequencing bias associated to GC-content and DNA sheering process (Park, 2009). We will refer to input-DNA as $x^{\text{input}} = \{x_1^{\text{input}}, \ldots, x_L^{\text{input}}\}$.

2.1.1 Read mappability filtering We ignore reads mapping to genomic regions which are either unassembled (denoted by Ns) or that exhibit a poor mappability. Poor mappability regions stem from the fact that short reads cannot be uniquely mapped to repetitive regions that exhibit a higher length than the reads themselves (Song and Smith, 2011). Reads aligned completely to a poor mappability region are ignored with the use of this filter.

2.1.2 Fragment size According to the ChIP-seq protocol, only the beginning of the sample's DNA fragments is sequenced (Park, 2009). To reconstruct the missing part of the DNA fragments, one has to compute the read fragment size f. Given the set F of the left most positions of all reads aligned to the forward strand and, respectively, R, the right most positions of all reads aligned to the reverse strand. We define the strand cross-correlation function $c(f) = \sum_{p \in F \cup R} h(p) \cdot h(f+p)$ following (Mammana *et al.*, 2013) with

$$h(p) = \begin{cases} 0, & p \notin F \cup R, \\ 2, & p \in F \cap R, \\ 1, & \text{else.} \end{cases}$$

The convolution c gives the correlation between counts on the forward and reverse strands for a given fragment size f. We are interested to find the fragment size $\hat{f} = \arg\max_{f \in F} c(f)$, that is the value with the maximum correlation between both strands.

2.1.3 Signal profile We extend all forward (reverse) reads from the leftmost (rightmost) position to the 3′ (5′) direction by the estimated read fragment size $\hat{f}$. We use a sliding window approach to partition the genome into a set $\{b_1, \ldots, b_L\}$.

Each bin b_j covers the genomic positions $[j \cdot s - 0.5 \cdot w, j \cdot s + 0.5 \cdot w]$, where s and w are the step size and the window size. The value of the genomic profile x_{ij} is simply the number of extended reads of ChIP-seq signal i aligned to regions overlapping bin b_j. If a read lies entirely in a filtered region (Section 2.1.1), it will be ignored.

2.1.4 GC-content Sequencing technologies usually exhibit an unwished correlation between the number of reads and the GC-content of the regions the reads come from (Benjamini and Speed, 2012). To model and correct this effect, we use an histogram-based approach introduced by Ashoor *et al.* (2013). Let g_j indicate the GC-content of the genomic bin b_j, that is the proportion of Gs and Cs in the bin's underlying sequence. We want to measure the average number of reads from input signal x^{input} assigned to bins on a particular GC-content interval. For an interval $[v, v+\delta]$ and genomic signal x^{input}, we have

$$h(v) = \frac{\sum_{j=1}^{L} x_j^{\text{input}} \mathbf{1}(g_j \in [v, v+\delta])}{\sum_{j=1}^{L} \mathbf{1}(g_j \in [v, v+\delta])},$$

where $\mathbf{1}$ is an indicator function and $v \in \{0.0, \ldots, 1-\delta\}$.

Moreover, we define the sum of average number of reads per GC-content $T = \delta \cdot \sum_{v=0} h(v)$. We correct the genomic signal x_{ij} with $g_j \in [v, v+\delta]$ as

$$x_{ij}^{\text{GC}} = x_{ij} \cdot \frac{T}{h(v)}.$$

Loosely speaking, we increase (decrease) the genomic signal of a bin, if the average GC-dependent signal is lower (higher) than expected.

2.1.5 Input subtraction To avoid bias associated to the DNA shearing process, it is usual to subtract the input-DNA from the ChIP-seq genomic signals. We follow the sequencing extraction scaling (SES) approach of Diaz *et al.* (2012), which performs a signal normalization previous to the subtraction. The rationale is that while input-DNA- and ChIP-seq libraries usually have similar number of reads, the mass of ChIP-seq reads are concentrated in protein–DNA interaction sites. Therefore, a simple subtraction tends to over-penalize the ChIP-seq signal. For a scaling factor α^{SES} (see Diaz *et al.* (2012) for details) and input-DNA x^{input}, we perform for both signals

$$x_{i\cdot}^{\text{SES}} = x_{i\cdot} - \alpha \cdot x_i^{\text{input}}.$$

2.1.6 Normalization and bin filtering We perform a sequencing depth scaling approach to normalize the two ChIP-seq signals to be compared. Let $S^1 = \sum_{j=1}^{L} x_{1j}$, $S^2 = \sum_{j=1}^{L} x_{2j}$ be the signal's total sum for $x_{1\cdot}$ and $x_{2\cdot}$. We scale up the genomic signal with less overall signal by the factor $\gamma = \max(S^1/S^2, S^2/S^1)$. For example, if $S^1 < S^2$, we have

$$x_{1\cdot}^{\text{norm}} = \gamma \cdot x_{1\cdot}$$

with $\gamma = S^2/S^1$. We round all values to obtain count data again. For simplicity we refer to this signal as read counts in the following text. We filter all bins b_j with low number of reads $(x_{1j} + x_{2j} > 3)$ and $\frac{x_{1j}}{S^1} + \frac{x_{2j}}{S^2} < \frac{2}{L \cdot M}$, where M represents the proportion of mappable regions. This filtering steps discards 80% of bins with low count evidence.

2.2 DPC HMM

Our HMM receives as input a matrix $\mathbf{X}$ with two genomic signals representing the ChIP-seq values of two biological conditions to be compared. The signals are obtained after the application of all pre-processing steps described in Section 2.1. We define a first-order HMM containing a state for DPs gained in signal $x_{1\cdot}$ (`Gain 1`), a state for DPs gained in signal $x_{2\cdot}$ (`Gain 2`) and a background state (`Back`), as depicted in Figure 2. All states have transitions to other states and to themselves. We use the product of Binomial or mixture of Poisson distributions as emissions for our HMM. More formally, for a given state s, we have $\Pr_s(x_{\cdot j}) = \Pr_{s1}(x_{1j}) \cdot \Pr_{s2}(x_{2j})$. In case of the Binomial distribution as emission, we have

$$\Pr_{si}(x_{ij}) = \mathrm{B}(x_{ij} = k \mid n, p_{si}) = \binom{n}{s} p_{si}^k (1 - p_{si})^{n-k},$$

where n represents the number of events observed (number of reads in the libraries) and p_{si} is the probability of observing a read in state s and signal i.

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

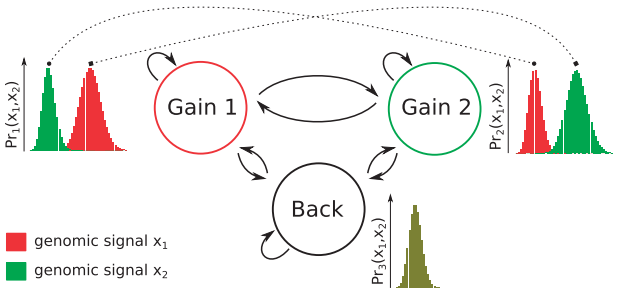

Fig. 2. Here we depict the three state HMM used for DP calling and samples of their emission distributions. We constraint the emission distribution of state `Back` as well as the emission distributions linked by dotted lines

In contrary to the Negative Binomial, the Binomial distribution is not able to model over-dispersion (Leleu *et al.*, 2010). However, there is no closed form to obtain maximum likelihood estimates of these densities and it would require numerically intricate extensions of the Baum–Welch algorithm. Alternatively, we evaluate the use of a mixture of Poisson distribution, which has a close form and can cope with over-dispersion. In this case, we have

$$\Pr_{sil}(x_{ij}) = \sum_{l=1}^{M} c_{sl} \cdot b_{sil}(x_{ij}).$$

where M is the number of mixture components, c_{sl} are the mixing coefficients and

$$b_{sil}(x_{ij}) = \frac{\exp(-f(l) \cdot \Theta_{si1}) \cdot (f(l) \cdot \Theta_{si1})^{x_{ij}}}{x_{ij}!}.$$

We use function $f(l) = l$ to ensure that the mean of each mixture component are multiple of each other (see Supplementary Section 2 for more details). This constraint was introduced to mitigate the fact that during mixture estimation some components can end up with little data support (or low mixing coefficients). This is usual when outliers (peaks with unusually high number of reads) are present in the data.

2.2.1 Initial HMM parameter We use a simple fold criteria to define a set of DPs to be used for the HMM initialization. More formally, a bin b_j will be assigned to state `Gain 1` if $x_{1j}/x_{2j} > t$, to state `Gain 2` if $x_{2j}/x_{1j} > t$ and to `Back` state otherwise. We use this annotation to define a posterior probability and to perform a single M-Step of the Baum–Welch algorithm to obtain initial parameters. Given the large size of the genomic signals, we only use a random selection of regions to train the HMM. We select genomic regions formed by contiguous bins not filtered in Section 2.1.1, which have at least a bin annotated with either `Gain 1` or `Gain 2` state. The selection is done until the training set has at least 3×10^6 bins.

2.2.2 HMM training The HMM is estimated with the Baum–Welch algorithm (Rabiner, 1989). Estimates of the initial state and transition probabilities follow usual methods. Concerning the emission distributions, the parameter n is equal in emissions and represents the number of reads of the largest library $n = \max(S^1, S^2)$ (the normalization steps ensure $S^1 \approx S^2$).

To reduce the number of parameter estimates, we constrain the parameters from `Back` state ($s = 3$) to be equal $p_{31} = p_{32}$. We also constrain emissions for state `Gain 1` (s = 1) and state `Gain 2` (s = 2) by $p_{11} = p_{22}$ and $p_{12} = p_{21}$. This makes the distributions of enriched signals (non-enriched signals) from states `Gain 1` and `Gain 2` to be equal (Fig. 2). For the Binomial distribution, given r_{sj} to be the posterior probability that the HMM is at state s after observing $x_{\cdot j}$, it can be shown that the maximum likelihood estimates of those parameters are:

$$p_{11} = p_{22} = \sum_{j=1}^{N} \frac{r_{1j}x_{1j} + r_{2j}x_{2j}}{n \cdot r_{1j} + n \cdot r_{2j}}$$

$$p_{12} = p_{21} = \sum_{j=1}^{N} \frac{r_{1j}x_{2j} + r_{2j}x_{1j}}{n \cdot r_{1j} + n \cdot r_{2j}}$$

$$p_{31} = p_{32} = \sum_{j=1}^{N} \frac{r_{3j}x_{1j} + r_{3j}x_{2j}}{2 \cdot n \cdot r_{3j}}$$

For the mixture of Poisson distribution, we can compute $_{si1}$ for the first component as

$$\Theta_{si1} = \frac{\sum_{j=1}^{L}\sum_{l=1}^{M} x_{ij} \cdot \gamma_{sl}(j)}{\sum_{j=1}^{L}\sum_{l=1}^{M} f(l) \cdot \gamma_{sl}(j)},$$

and for component l

$$\Theta_{sil} = l \cdot \Theta_{si1}$$

where $\gamma_{sl}(j)$ is the posterior probability of being in component l at state s at time j. All other parameters follow standard mixture model estimates. We constrain the HMM's mixture distributions accordingly to the case of the Binomial distribution (see Supplementary Section S2 for more details of the mixture of Poisson). Training is performed until convergence. Finally, we apply the Viterbi algorithm to the complete genomic signals. We merge all consecutive bins visited by either states `Gain 1` or `Gain 2` on the Viterbi path to obtain the candidate DPs.

2.3 Post-processing

Next, we perform two post-processing steps to remove spurious DPs. First, we ignore all DPs with a size smaller than the estimated fragment size $\hat{f}$. We also merge concordant DPs, which have a distance less than the estimated fragment size $\hat{f}$. The second step is only suggested for histone modification data, which are usually localized in broader genomic regions.

We perform a statistical test to assign a P value to each DP. Let $y_1 = \sum_{j=u}^{v} x_{1j}$ and $y_2 = \sum_{j=u}^{v} x_{2j}$ be the read counts for a DP spanning from bins u to v. For DP `Gain 1`, the P value is the sum of probabilities of the tuple (a,b) such that $a > y_1$, $a + b = y_1 + y_2$. More formally,

$$\Pr(a > y_1 | y_2) = \sum_{a > y_1;\ a+b=y_1+y_2} \Pr(a, b),$$

where $a, b \in \mathbb{N}$. We compute the probability $\Pr(a, b)$ as

$$\Pr(a, b) = \frac{\mathrm{B}(a \mid n, p_{31}) \cdot \mathrm{B}(b \mid n, p_{32})}{\sum_{c+d=a+b} \mathrm{B}(c \mid n, p_{31}) \cdot \mathrm{B}(d \mid n, p_{32})},$$

where $c, d \in \mathbb{N}$ with $c + d = a + b$ and n, p_{31}, p_{32} are the parameters of the emission distributions of the `Back` state. The P value for DP `Gain 2` can be defined accordingly.

For large y_1, y_2 values, the computation of the above equations are computationally expensive. Following Anders and Huber (2010), we can combine both equation and obtain

$$\Pr(a > y_1 | y_2) = \frac{\sum_{a > y_1;\ a+b=y_1+y_2} \mathrm{B}(a \mid n, p_{31}) \cdot \mathrm{B}(b \mid n, p_{32})}{\sum_{c+d=y_1+y_2} \mathrm{B}(c \mid n, p_{31}) \cdot \mathrm{B}(d \mid n, p_{32})}.$$

The sum of the nominator is a subset of the sum of the denominator. Consequently, we only need to evaluate the sum of the denominator and

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

take into account the appropriate values for the nominator. Moreover, we can rewrite the main term in the nominator (denominator) as a function $f(a) = \mathrm{B}(a \mid n, p_{31}) \cdot \mathrm{B}(y_1 + y_2 - a, p_{32})$ given that $b = y_1 + y_2 - a$. This function is axially symmetrical and has a global maximum at $a_{\max} = (y_1 + y_2)/2$ given that $p_{31} = p_{32}$. That is, we only have to evaluate half of the sum's values of the numerator (denominator). Furthermore, as the function decrease monotonically departing from $a_{\max}$, we can approximate the P value calculation by making $f(e) = f(a)$ for all $e > a$ given that $f(a) - f(a+1) < \epsilon$. These steps allow a speed up of 100 times on the P value calculations on our experiments. Note that these improvements would not be possible in the mixture of Poisson distribution, therefore, we estimate a Binomial distribution using the posterior probability of the background model in this case.

3 EXPERIMENTAL DESIGN

3.1 Real datasets

We used data from two studies, both using cells of the mouse immune system, to evaluate the performance of our method. The first study analyses the response of macrophages after activation of the TLR4 signaling pathway (Kaikkonen *et al*., 2013). We use ChIP-seq experiments from the TF PU.1 at time points 0, 1, 6, 12 and 24 h and from the histone modification H3K4me2 at time points 0, 1, 6 and 24 h (time point 12 h was not available). We perform DPC by comparing the time point 0 h with all other time points which leads to seven experiments (Table 1). The study provides an input-DNA signal of untreated cells (0 h), which is used as control. Moreover, we also use the genomic run-on sequencing (GRO-seq) experiments, which measures the quantity of nascent transcripts, in time points 0, 1, 6, 12 and 24 h for validation. These data were obtained from GEO accession number GSE48759.

The second study comprises in-house data describing regulatory changes during the development of antigen-presenting dendritic cells (DCs). Our institute has established an *in vitro* protocol to differentiate multi-potent progenitors (MPP) from adult mouse bone marrow to common DC progenitors (CDP; Felker *et al*., 2010). The CDP further differentiate into two distinct DC subsets, classical DC (cDC) and plasmacytoid DC (pDC). These four cell types were sorted by FACS and ChIP-seq experiments for H3K4me1 and PU.1 were performed (Lin *et al*., in preparation, data available at GSE57563). We have performed a DP analysis comparing the lineage commitment steps (MPP to CDP, CDP to cDC, CDP to pDC) and DC subset specification (cDC and pDC). This leads to eight further experiments as listed in Table 1. We have gene expression data from microarrays for these four cell types from Felker *et al*. (2010) (GEO accession GSE22432).

We use BWA (Li and Durbin, 2010) version 0.6.1 − r104 with standard parameter for read mapping. All experiments were based on mouse genome (mm9).

3.2 Evaluation

There is no gold standard to evaluate DPs. A strategy commonly used to evaluate SPCs applied to ChIP-seq from TFs—measuring the proportion of candidate peaks with motifs of the TF (Chen *et al*., 2012)—is not applicable to the DP problem. Sequence evidences (motifs) are independent of the cellular context and they will be found in the genomic sequence if binding is present in at least one of the cellular conditions. An alternative is to associate changes in protein–DNA with changes in gene expression, whenever gene expression is measured in the same cellular conditions, as usually performed in genomic studies with ChIP-seq data (Heinz *et al*., 2010; Kaikkonen *et al*., 2013).

Table 1. Overview of DP experiments; TLR4 experiments regards data from Kaikkonen *et al*. (2013) and DC experiments from the dendritic cell development study (Lin *et al*., in preparation)

Exp.	Name	Protein	Signal 1	Signal 2
TLR4	PU.1-0h-1h	PU.1	0h	1h
TLR4	PU.1-0h-6h	PU.1	0h	6h
TLR4	PU.1-0h-12h	PU.1	0h	12h
TLR4	PU.1-0h-24h	PU.1	0h	24h
TLR4	H3K4me2-0h-1h	H3K4me2	0h	1h
TLR4	H3K4me2-0h-6h	H3K4me2	0h	6h
TLR4	H3K4me2-0h-24h	H3K4me2	0h	24h
DC	PU.1-MPP-CDP	PU.1	MPP	CDP
DC	PU.1-CDP-cDC	PU.1	CDP	cDC
DC	PU.1-CDP-pDC	PU.1	CDP	pDC
DC	PU.1-cDC-pDC	PU.1	cDC	pDC
DC	H3K4me1-MPP-CDP	H3K4me1	MPP	CDP
DC	H3K4me1-CDP-cDC	H3K4me1	CDP	cDC
DC	H3K4me1-CDP-pDC	H3K4me1	CDP	pDC
DC	H3K4me1-cDC-pDC	H3K4me1	cDC	pDC

Here, we propose a method that quantifies changes in expression in the proximity of DPs. The measure is independent of the number of called peaks and can be applied either to gene expression data from sequencing or microarray data. For sequencing data, we first extend the DPs to have at least a length of 1000 bps and then count the number of RNA/GRO-seq reads aligned to these regions for both cell samples. The use of small windows of RNA/GRO-seq is based on the fact that we want to capture the expression of known genes or uncharacterized lncRNAs in the close proximity of the DP. See Supplementary Figure 7 for an example of GRO-seq profiles.

More formally, we sort by increasing P value and take the top k ranked DPs for `Gain 1` (or `Gain 2`). Next, we evaluate the logarithmic ratio $\log(s_1/s_2)$ for the top k DPs, where s_1 and s_2 are the RNA/GRO-seq read counts associated to peaks in the first and second condition. By increasing k, we obtain curves depicted in Figure 4. The higher the value, the higher is the association between DP and changes in expression.

Regarding gene expression from microarrays, we need an additional step to associate peaks to genes. Peaks are assigned to genes if they are located inside the gene body or close to the gene's promoter (1000 bp upstream); or if the peaks are located 50 Kbps away from the TSS and there is no other gene's TSS in between. The average expression value of genes assigned to a peak is used. Peaks not assigned to genes are ignored for the statistics.

We compute the integral of the curves obtaining the single statistic differential average gene expression (DAGE). Let $e(k)$ be the average log ratio expression associated to the best k DPs

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

for either `Gain 1` or `Gain 2`. We define

$$\mathrm{DAGE} = \sum_{k=h}^{H} |e(k)| \cdot h.$$

for $k \in [h, 2 \cdot h, \ldots, H]$ where h is the step size and H the maximum number of DPs used. For visualization purposes, we display the negative values of the DAGE curve for DP `Gain 2`.

3.3 Simulated datasets

The simulation of single ChIP-seq datasets has already been addressed by Zhang *et al.* (2008) and Humburg (2011), but none of these approaches can be directly used in the DPC problem. We, therefore, developed an algorithm inspired by Humburg (2011) to generate ChIP-seq reads simulating a pair of biological conditions with DPs. In short, for a given reference genome the procedure works as follows: (1) selecting genomic regions to include protein domains (region with several binding proteins) and sampling the number of proteins in a domain following a negative Binomial distribution NB_{m_1,p_1}; (2) sampling and placement of reads per protein following a Negative Binomial distribution NB_{m_2,p_2} and (3) assignment of a proportion of reads from a protein to each of the simulated biological condition. We use the mean and standard deviation to describe the Negative Binomial distribution. We take into account the original position of the proteins and the proportion of reads to define the true DP (see Supplementary Section S1 for details).

We are particularly interested in the effect of protein domain sizes as well as the number of reads in the libraries. We set the parameters (m_1,p_1) to $\{(1, 4), (4, 6), (8, 14)\}$ obtaining datasets with an increasing number of proteins per domain. Furthermore, we set the parameters (m_2,p_2) to $\{(20, 200), (20, 2000), (100, 200)\}$ obtaining both datasets with distinct number of reads per peak as well as high variance of reads per peak. We use chromosome 1 of the mouse genome (mm9) as reference genome. For each parameterization choice, we generate 50 pairs of simulated datasets. Methods were evaluated by sorting the DPs by smallest P value and calculating the proportion of true positives among the top r called DPs. A candidate DP is considered a true positive if it overlaps with a true DP (see Supplementary Section S1 for more details).

3.4 HMM parameterization

For all datasets, we compute the fragment size from the range $\hat{f} = \arg\max_{f \in F} c(f)$, for $F = [0, 5, \ldots, 600]$. Moreover, we use a step size of 50 and window size of 100 to compute the signal profile. This choice was based on visual inspection of peaks: smaller windows did not affect peaks and larger windows induced too large peaks. We only use input-DNA signal of chromosome 1 to build the GC-content histogram. We use the mappability files that are provided by Landt *et al.* (2012) (http://hgdownload-test.cse.ucsc.edu/goldenPath/mm9/encodeDCC/wgEncodeMapability/, last access: 20th March 2014). We only consider regions with an mappability value of 1. We implemented our HMM in Python 2.7 by using the SciKit package (Pedregosa *et al.*, 2011). ODIN is part of the Regulatory Genomics Toolbox (see www.regulatory-genomics.org). Predictions for experiments and simulated data are available at the Web Supplement http://costalab.org/wp/odin. In our experiments, ODIN (with a single component in the mixture model) requires on average 12GB of memory. The calculations last on average 4 h on a 2.4 GHz machine. Computational time increases linearly with the number of components in the case of mixture models.

3.5 Competing methods

We use PeakSeq, Quest and MACS as SPCs in combination with DBChIP, DESeq and MAnorm. The SPCs were selected based on their good performance (Chen *et al.*, 2012; Wilbanks and Facciotti, 2010). We run all SPCs with standard parameter and corresponding input-DNA. The simulated data does not provide input-DNA. As QUEST and PeakSeq can only be executed with input signal, they are not evaluated in the simulation experiments. We refer to the obtained peaks as candidate peaks for the two-stage DPC. DBChIP is proposed for TF data only. Therefore, it uses a fixed window size of 250 bps around the summits of candidate peaks. We also define an own two-stage DPC. This approach merges all candidate peaks and uses them as input for DESeq. Note that distinctly from DBChIP, this approach can find DPs with variable size common in histone ChIP-seq. We separately run DBChIP, DESeq, MAnorm with default parameters and the candidate peaks of PeakSeq, Quest and MACS.

ChIPDiff is also run with default parameters ($FC = 3$, minRegionDist = 1000 and minP = 0.95). It finds less than 20 peaks for half of the experiments from the TLR4 study. For these experiments, we change parameters (FC = 1.5, minRegionDist = 200 and minP = 0.7) to obtain at least 100 DPs. Moreover, as ChIPDiff does not provide P values or any criteria for sorting peaks, we can only obtain points for the DAGE statistic. We also try to use RSEG with no success. For the dataset PU1-MPP-CDP, it returns very large peaks (mean size of 58 716 bps) in comparison to ODIN (315 bps). Per definition, RSEG is tailored for identifying broad genomic domains and will not be considered here. MACS2 is run with parameter $C = 0.5$ for the data from Kaikkonen *et al.* (2013) and $C = 1.5$ for the DC data.

We run DBChIP, MAnorm, ChIPDiff, MACS2, our DESeq approach and ODIN for all 15 experiments. DBChIP is only run for TFs data and our DESeq approach for histone data. Therefore, we distinguish between experiments with TF or histone modifications. We will call DPs to be `Gain 1` (`Gain 2`) for all competing methods, whenever they are detected to have higher signal in $x_{1\cdot}$ ($x_{2\cdot}$).

We then apply the Friedman-Nemenyi test (Demšar, 2006) on the DAGE values. This non-parametric test checks significant differences when more than two methods are applied to multiple datasets. It works in two steps: first, it ranks the methods under comparison and second, if a significant differences exist, it identifies the pairs of methods causing them. Here, the Friedman–Nemenyi test indicates whether one of the methods is assigned to significant higher DAGE values than others.

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

4 RESULTS

4.1 Genomic signal construction

We investigate the effect of the preprocessing steps to create the genomic signal (Section 2.1). In particular, we analyze all eight combinations of using: (1) the GC-content model, (2) filtering reads aligned to poor mappability regions and (3) the subtraction of input-DNA. DP predictions are based on chromosome 1 for all 15 experiments on real data. The Friedman–Nemenyi test on DAGE statistics for $h = 50$, $H = 500$ indicates a slight advantage of using input-DNA subtraction and GC-content model compared to using none of the steps for TF data (P value < 0.1). No significant difference is detected on histone data. However, the Friedman score ranks are similar in both scenarios reinforcing the advantage of the input-DNA subtraction and GC-content model, which will be further used in all experiments (see Supplementary Tables S3–S6).

4.2 Method parameterization

We empirically evaluate the use of parameter constraining and the choice of the HMM's emission distribution as presented in Section 2.2.2. Again, we restrict the analysis to the chromosome 1 of all 15 real data experiments and compute the DAGE statistic ($h = 50$, $H = 500$) with or without parameter constraining. The constraint model has statistical significant higher DAGE values (P value < 0.006, one-tailed Wilcoxon test) for experiments with TF, while no significant differences are obtained on histone modification experiments. This reinforces the advantage of parameter constraining, which is used in further experiments.

Furthermore, we evaluate the use of distinct distributions: Binomial and mixture of Poisson with 1–4 components. As shown in Supplementary Tables S7–S10, no significant difference was found. We, therefore, use the Poisson mixture with the number of components that offers the highest ranking (4 for histones and 1 TFs) as well as the Binomial distribution in the following experiments.

Moreover, we evaluate the P value estimation methods of ODIN, DESeq and EdgeR. We use DPs predicted by ODIN with Binomial distribution. ODIN's P value estimation has a significant higher DAGE score than DESeq and EdgeR for TF experiments and a significant higher DAGE score than EdgeR for histone experiments (see Supplementary Tables S15–S18).

Finally, we inspect the impact of SPCs (MACS, QUEST and PeakSeq) on two stage peak callers DPChIP, DESeq and MAnorm. As shown in Supplementary Tables S11–S14, significant difference between the use SPCs was found. We use the best ranked combination: MAnorm-macs, DESeq-quest and DBChIP-quest.

We discard chromosome 1 regions from all further analysis of real datasets, as they have been used for parameterization experiments.

4.3 Comparative analysis on simulated data

Supplementary Figure S1 shows the results for simulated data. As expected, methods obtained best performance on experiments with more reads and less number of proteins per domain (bottom left). The performance of MAnorm (green line) for top ranked DPs is quite competitive with ODIN variants (read and yellow lines) on data with few proteins per domains (left column), but its performance deteriorates whenever more proteins are present in the domains. MACS2 (blue line) has similar performance as ODIN variants (read and yellow lines) when large number of reads are present (bottom), but ODIN clearly outperforms MAC2 when peak sizes have a high variance (middle row). We calculate the area under the curve for each experiment to perform a Friedman–Nemenyi test. We then evaluate the overall performance of methods for all conditions (Supplementary Tables S1 and S2). Results indicates that ODIN with Binomial or single Poisson distribution has a significantly higher AUC than MACS2, MAnorm, DESeq and DBChIP.

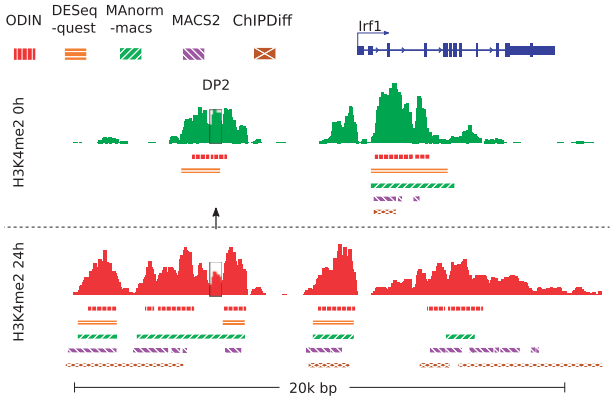

Fig. 3. DPs detected on experiment H3K4me2-0h-24h around the Irf1 gene. Bars bellow the ChIP-seq signal indicate the regions called as DPs by distinct methods

4.4 Comparative evaluation on real data

We perform a visual inspection of the DPs from experiment H3K4me2-0h-24h around the Irf1 gene (Fig. 3). MAnorm and our DESeq approach, which have same predictions for this region, can successfully detect changes in large peak areas. ChIPDiff detects most DPs, but have a tendency to call large regions. ODIN and MACS2 are able to detect detailed changes within the large domains. Note that MACS2 and ChIPDiff are not able to recover a DP upstream of Irf1 (marked as DP2) on H3K4me2 0h. The lost of this histone mark after TLR4 treatment is supported by gain of a PU.1 on the very sample location (see Supplementary Fig. S7 for PU.1 ChIP-seq profiles).

In Figure 4, we display DAGE curves for DBChIP, MACS2, our DESeq approach, MAnorm and ODIN for four selected experiments on real data. As ChIPDiff does not provide information to sort the DPs, its results are only represented as points, where the x-axis location corresponds to the number of called DPs. In most scenarios, curves approximate to zero for higher ranks, which indicates that higher ranked DPs are associated to higher expression changes. In some scenarios, such as H3K4me2-0h-6h, the curve associated to `Gain 2` peaks (right bottom) are further from 0 than `Gain 1` peaks (right middle). This is an indication that there are more changes in ChIP-seq peaks and gene expression in signal 2 (6 h) than in signal 1 (0 h). This is in accordance with the main message of the TLR4 study, which shows that induction of TLR4 promotes new enhancers

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

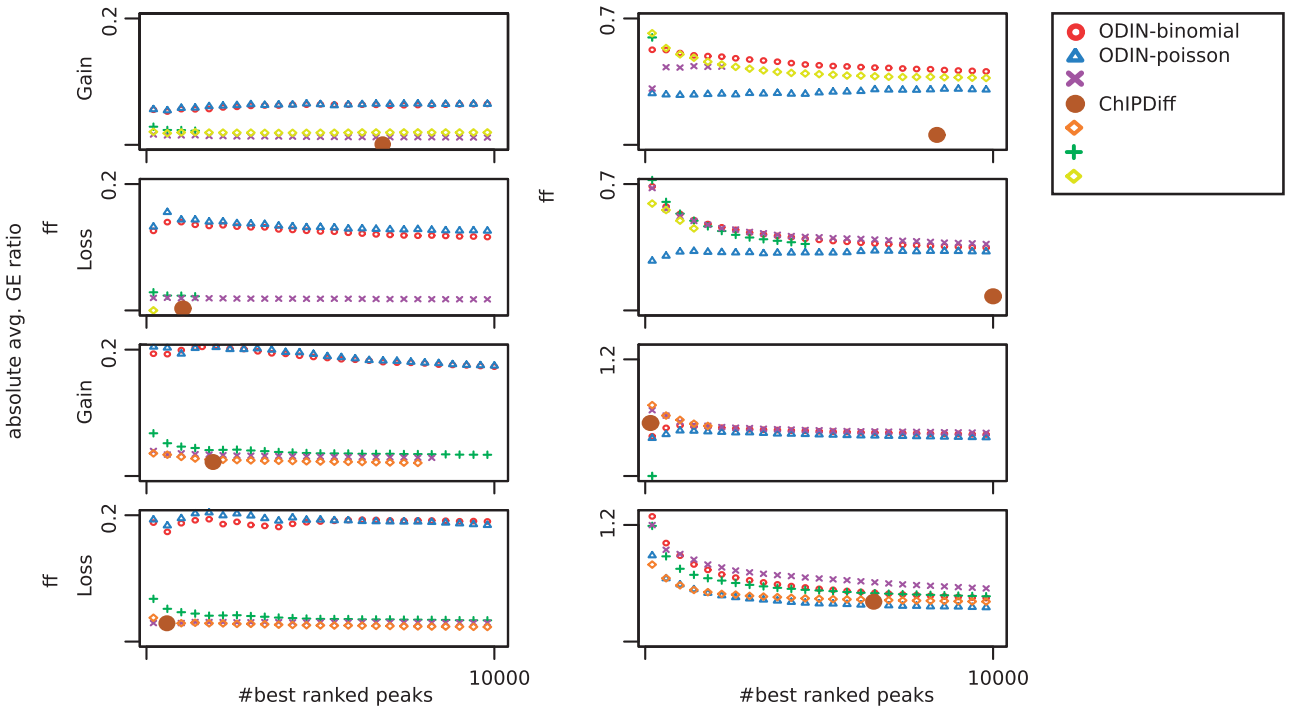

Fig. 4. Here we depict the DAGE curves for selected experiments from TLR4 and DC studies. Lines in the first and third row represent DP gained in the first signal (`Gain 1`), while lines in the second and fourth row in the second signal (`Gain 2`)

marked by H3K4me2 (Kaikkonen *et al.*, 2013). While there are some experiment-specific variations, both ODIN variants outperforms other methods on PU.1-MPP-CDP, H3K4me1-MPP-CDP, while the performance of ODIN with Binomial distribution is similar to other methods on H3K4me1-0h-6h and PU.1-0h-6h. All DAGE curves can be found at Supplementary Figures S3–S6.

Finally, we evaluate the performance of all methods over the 15 real data experiments listed in Table 1 by computing the DAGE scores for Gain 1 and Gain 2 peaks. The Friedman–Nemenyi test indicates that both ODIN variants have significantly higher DAGE scores than DBChIP and MACS2 on TF experiments (P value <0.1, see Supplementary Tables S19 and S20) and significantly higher DAGE scores than DESeq on histone data (P value <0.05, see Supplementary Tables S21 and S22). Moreover, ODIN with Binomial distribution has a significant higher DAGE score than MACS2 and MAnorm on histone data. We also performed an evaluation of ChIPDiff by comparing the DAGE values of all methods with H equal to the number of peaks called by ChIPDiff. ODIN with Binomial distribution has significantly higher DAGE scores than ChIPDiff on TF experiments (P value <0.1, see Supplementary Table S24), while no statistical difference was detected on histone data (see Supplementary Table S26). In all cases, both ODIN with binomial and mixture of Poisson distribution ranked best by the Friedman score compared to all competing methods.

5 CONCLUSION AND FUTURE WORK

We propose ODIN, a DPC method that performs genomic signal processing, DPC and P value calculation in an integrated framework. Empirical analysis shows that DPs detected by ODIN are best associated to changes in gene expression of genes neighboring these DPs than several of its competing methods. ODIN can be used with either a Binomial distribution or a mixture of Poisson distribution. It outperforms methods using complex distributions as the Negative Binomial used by DBChIP and PeakDE. These results are further supported with simulated data, where ODIN outperforms competing methods on scenarios with few reads and complex DPs. Moreover, we present the first approach to evaluate DPs methods, which is based on the association of DPs with expression changes in the same cellular conditions; and a methodology to simulate pairs of ChIP-seq read libraries with DPs.

Calling DPs is an extremely important but so far poorly explored problem of ChIP-seq analysis. Epigenomic consortia are planning the analysis of ChIP-seq of patients with distinct disease phenotypes (Adams *et al.*, 2012). Methods capable for detection of DPs by simultaneous analysis of several ChIP-seq data will be crucial for such medical epigenomics applications. Aspects such as the presence the genetic variability of patients (Ashoor *et al.*, 2013) and the over-dispersion of peaks sizes in biological replicates are open challenges. We plan to extend ODIN to allow the analysis of biological replicates and the simultaneous analysis of more than two cellular conditions.

Funding: This work was supported by the Interdisciplinary Center for Clinical Research (IZKF Aachen), RWTH Aachen University Medical School, Aachen, Germany; the Excellence Initiative of the German Federal and State Governments and the German Research Foundation through Grant GSC 111;

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

and the START-program (AZ 22/13) of the Faculty of Medicine, RWTH Aachen. Part of the work was funded by the donation of U. Lehmann and the German Research Foundation (DFG ZE432/5-2) to M.Z.

Conflict of interest: none declared.

REFERENCES

Adams,D. *et al.* (2012) BLUEPRINT to decode the epigenetic signature written in blood. *Nat. Biotechnol.*, **30**, 224–226.

Anders,S. and Huber,W. (2010) Differential expression analysis for sequence count data. *Genome Biol.*, **11**, R106+.

Ashoor,H. *et al.* (2013) HMCan: a method for detecting chromatin modifications in cancer samples using ChIP-seq data. *Bioinformatics*, **29**, 2979–2986.

Benjamini,Y. and Speed,T.P. (2012) Summarizing and correcting the GC content bias in high-throughput sequencing. *Nucleic Acids Res.*, **40**, e72.

Chen,Y. *et al.* (2012) Systematic evaluation of factors influencing chip-seq fidelity. *Nat. Methods*, **6**, 609614.

Dahl,J.A. and Collas,P. (2008) MicroChIP–a rapid micro chromatin immunoprecipitation assay for small cell samples and biopsies. *Nucleic Acids Res.*, **36**, e15.

Demšar,J. (2006) Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, **7**, 1–30.

Diaz,A. *et al.* (2012) Normalization, bias correction, and peak calling for ChIP-seq. *Stat. Appl. Genet. Mol. Biol.*, **11** (3), Article 9.

Felker,P. *et al.* (2010) Tgf-beta1 accelerates dendritic cell differentiation from common dendritic cell progenitors and directs subset specification toward conventional dendritic cells. *J. Immunol.*, **185**, 5326–5335.

Heinz,S. *et al.* (2010) Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. *Mol. Cell*, **38**, 576–589.

Humburg,P. (2011) *ChIPsim: Simulation of ChIP-seq experiments*. R package version 1.18.0.

Kaikkonen,M.U. *et al.* (2013) Remodeling of the enhancer landscape during macrophage activation Is coupled to enhancer transcription. *Mol. Cell*, **51**, 310–325.

Kuan,P.F. *et al.* (2011) A Statistical framework for the analysis of ChIP-Seq data. *J. Am. Stat. Assoc.*, **106**, 891–903.

Landt,S.G. *et al.* (2012) ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia. *Genome Res.*, **22** (9), 1813–1831.

Leleu,M., Lefebvre,G. and Rougemont,J. (2010) Processing and analyzing chip-seq data: from short reads to regulatory interactions. *Brief. Funct. Genom.*, **9**, 466–476.

Li,H. and Durbin,R. (2010) Fast and accurate long-read alignment with BurrowsWheeler transform. *Bioinformatics*, **26**, 589–595.

Liang,K. and Keles,S. (2012) Detecting differential binding of transcription factors with ChIP-seq. *Bioinformatics*, **28**, 121–122.

Lin,Q. *et al.* (2014) Dynamic chromatin signatures and cis-regulatory network control dendritic cell development. *In preparation*, .

Mammana,A., Vingron,M. and Chung,H.-R. (2013) Inferring nucleosome positions with their histone mark annotation from chip data. *Bioinformatics*, **29**, 2547–2554.

Martens,J.H.A. and Stunnenberg,H.G. (2013) BLUEPRINT: mapping human blood cell epigenomes. *Haematologica*, **98**, 1487–1489.

Maston,G.A. *et al.* (2006) Transcriptional regulatory elements in the human genome. *Annu. Rev. Genom. Hum. Genet.*, **7**, 29–59.

Park,P.J. (2009) ChIP-seq: advantages and challenges of a maturing technology. *Nat. Rev. Genet.*, **10**, 669–680.

Pedregosa,F. *et al.* (2011) Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.*, **12**, 2825–2830.

Rabiner,L.R. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77**, 257–286.

Robinson,M.D., McCarthy,D.J. and Smyth,G.K. (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**, 139–140.

Rozowsky,J. *et al.* (2009) PeakSeq enables systematic scoring of ChIP-seq experiments relative to controls. *Nat. Biotechnol.*, **27**, 66–75.

Shao,Z. *et al.* (2012) MAnorm: a robust model for quantitative comparison of ChIP-Seq data sets. *Genome Biol.*, **13**, R16+.

Song,Q. and Smith,A.D. (2011) Identifying dispersed epigenomic domains from ChIP-Seq data. *Bioinformatics*, **27**, 870–871.

Spyrou,C. *et al.* (2009) Bayespeak: Bayesian analysis of chip-seq data. *BMC Bioinformatics*, **10**, 299.

Wilbanks,E.G. and Facciotti,M.T. (2010) Evaluation of algorithm performance in ChIP-seq peak detection. *PloS one*, **5**, e11471+.

Xu,H. *et al.* (2008) An hmm approach to genome-wide identification of differential histone modification sites from chip-seq data. *Bioinformatics*, **24**, 2344–2349.

Zhang,Z.D. *et al.* (2008) Modeling chip sequencing in silico with applications. *PLoS Comput. Biol.*, **4**, e1000158.

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016