entropy

MDPI

Article

A Clustering Approach for Motif Discovery in ChIP-Seq Dataset

Chun-xiao Sun [1], Yu Yang [2,3,*], Hua Wang [4,5,*] and Wen-hu Wang [2]

[1] College of Science, Northwest A&F University, Yangling 712100, China
[2] School of Computer Science, Pingdingshan University, Pingdingshan 467000, China
[3] School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai 200240, China
[4] College of Software, Nankai University, Tianjin 300071, China
[5] Department of Mathematical Sciences, Georgia Southern University, Statesboro, GA 30460, USA
* Correspondence: yangyu@sjtu.edu.cn (Y.Y.); hwang@georgiasouthern.edu (H.W.)

Received: 6 June 2019; Accepted: 15 August 2019; Published: 16 August 2019

Abstract: Chromatin immunoprecipitation combined with next-generation sequencing (ChIP-Seq) technology has enabled the identification of transcription factor binding sites (TFBSs) on a genome-wide scale. To effectively and efficiently discover TFBSs in the thousand or more DNA sequences generated by a ChIP-Seq data set, we propose a new algorithm named AP-ChIP. First, we set two thresholds based on probabilistic analysis to construct and further filter the cluster subsets. Then, we use Affinity Propagation (AP) clustering on the candidate cluster subsets to find the potential motifs. Experimental results on simulated data show that the AP-ChIP algorithm is able to make an almost accurate prediction of TFBSs in a reasonable time. Also, the validity of the AP-ChIP algorithm is tested on a real ChIP-Seq data set.

Keywords: motif discovery; transcription factor binding sites; planted motif search; ChIP-Seq

1. Introduction

Transcription factor binding sites (TFBSs) [1] are short, degenerated nucleotide fragments (usually $\leq$30 bps) located in specific DNA regions. They play an important role in regulating gene expression. The Planted (l, d) Motif Search (PMS) problem [2] is a popular motif model for the identification of TFBSs (i.e., motif discovery) in bioinformatics, and is formally defined as follows:

Definition 1 (PMS)**.** *Given a set of DNA sequences $X = \{X_1, X_2, \ldots, X_t\}$ with $|X_i| = n$ and three non-negative integers d, q, l, with $0 \leq d < l < n, 0 < q \leq t$, the PMS problem is to find an l-mer M (a string of length l), such that each selected sequence X_i has an l-mer M_i with Hamming distance $d_H(M, M_i) \leq d$, for $i = 1, 2, \ldots, q$. The l-mer M is called an (l, d) motif and the l-mer M_i is called a motif instance.*

According to the different values of q representing the distribution of motif instances, there are three different motif discovery sequence models [3]: (i) Exactly one motif occurrence per sequence (the OOPS model), (ii) zero or one motif occurrences per sequence (the ZOOPS model), or (iii) zero or more motif occurrences per sequence (the TCM model). For the OOPS model, $q = t$, for the ZOOPS model and TCM model, $0 < q < t$.

Generally, there are two kinds of algorithms for solving the PMS problem: exact algorithms and approximate algorithms. Exact algorithms [4–10] always use consensus sequences [11] to represent motifs and can find all (l, d) motifs. Most exact algorithms are pattern-driven algorithms, which attempt to enumerate all possible 4^l l-mers (substring patterns of length l) to find the l-mer with the maximum number of approximate occurrences. Approximate algorithms [12–18] usually use a position weight matrix (PWM) [19] to describe the most likely occurring motifs and can report results in a short time, but do not always identify all (l, d) motifs. Most approximate algorithms use probabilistic analysis to maximize the score function which describes how likely it is for an l-mer pattern to be a motif instance.

Recently, chromatin immunoprecipitation combined with next-generation sequencing (ChIP-Seq) technology has produced extremely valuable information for the genome-wide identification of transcription factor binding sites (TFBSs) and in the field of epigenetics, which mainly focus on DNA methylation, nucleosome localization, and histone modification. For transcription factors, ChIP-Seq is widely used to study the binding of transcription factors for the analysis of gene expression regulation on a genome-wide scale. For histones, ChIP-Seq performs high-throughput histone modification sequencing in the whole genome with sufficient sequencing depth and range, which not only improves the sensitivity and specificity of sequencing, but can also transform qualitative sequencing methods into quantitative detection.

In this paper, we focus on an algorithm to discover transcription factor binding sites in a ChIP-Seq data set. A ChIP-Seq data set is a set of peak regions containing TFBSs obtained through ChIP-Seq experiments, read mapping, and peak calling. It contains hundreds or more DNA sequences, which increases the difficulty of accurate and efficient identification of TFBSs.

Some algorithms have recently been proposed to discover TFBSs in ChIP-Seq data sets [20–29]. However, none of them has proven to be absolutely superior, compared to the rest. Some of these are tailored versions of previous motif discovery algorithms, specifically tailored towards ChIP-Seq data sets, such as MEME-ChIP [20] and HMS [21]. MEME-ChIP [20], which incorporates two complementary motif discovery algorithms, known as MEME and DREME [22], can identify motifs without restriction on the size or number of sequences, allowing very large ChIP-Seq data sets to be analyzed. HMS [21], which is an improved version of Gibbs Sampler, combines stochastic sampling and a deterministic greedy search step, which improves computation efficiency. DREME [22] is specifically designed to find short, core DNA-binding motifs of eukaryotic TFs, and is optimized to analyze very large ChIP-Seq data sets in just minutes. One may speed up the existing motif discovery algorithms by integrating some information, such as in the cases of STEME [23] and ChIP-Munk [24]. STEME [23] accelerates MEME by indexing sequences with suffix trees. ChIP-Munk [24] combines a greedy approach with an expectation-maximization (EM) algorithm to achieve a high efficiency. There are also exhaustive methods for determining exact motifs in ChIP-Seq data sets, such as FMotif [25] and Weeder [26]. FMotif [25] first constructs a mismatched suffix tree to scan and count all possible motif instances, and then implements a depth-first search to enumerate all possible motifs. However, the run time of FMotif becomes unrealistic with increasing values of l and d. Others use word enumeration methods to process full-size ChIP-Seq data sets, such as CisFinder [28] and MCES [29]. CisFinder [28] employs a word clustering method to group short l-mers ($l = 7, 8$, or 9), but struggles to find exact (l, d) motifs with larger values of l and d in ChIP-Seq data sets. MCES [29], a new planted (l, d) motif discovery algorithm, mines and combines substrings to rapidly identify exact motifs in full-size ChIP-Seq data sets.

In this paper, we propose a new motif discovery algorithm, named AP-ChIP, which is specially designed for better discovering TFBSs in ChIP-Seq data sets. The algorithm first constructs and then further filters cluster subsets using probabilistic analysis. Then, Affinity Propagation (AP) clustering [30] is applied to the candidate cluster subsets in order to discover optimal motifs. Experimental results show that the AP-ChIP Algorithm 1 can find TFBSs in a ChIP-Seq data set very efficiently and effectively.

2. Method

A ChIP-Seq data set has the following fundamental features: (i) Some of the sequences may contain no motifs at all; and (ii) thousands of sequences lead to huge amount of background l-mers. To cater to ChIP-Seq data sets, we design the AP-ChIP Algorithm 1 under the ZOOPS model and set some proper thresholds to filter redundant background l-mers. More specifically, the AP-ChIP Algorithm 1 consists of the following three steps:

2.1. Construct Cluster Subsets

We introduce the observation that any two motif instances x_1 and x_2, each of which differs from the same motif x up to d positions, must have Hamming distance of no more than $2d$, denoted as $d_H(x_1, x_2) \leq 2d$. Consequently, if an l-mer in one sequence is a motif instance, all other motif instances in the remaining sequences will be gathered in the corresponding cluster subset. Under the ZOOPS model, we choose h $(h = t - q + 1)$ sequences as the reference sequences to ensure that at least one sequence among the h sequences contains a motif instance [31]. In general, we use the first h sequences $\{X_1, X_2, \ldots, X_h\}$ as the reference sequences. As the l-mer which is the motif instance is not known in advance, we consider all l-mers $x_{i,j}$ $(i = 1, 2, \ldots, h; j = 1, 2, \ldots, n - l + 1)$ in the first h sequences as the reference subsequences.

The ideal cluster subsets are expected to contain as few background l-mers as possible and, also, to include sufficient motif instances. Therefore, we set a threshold k $(d < k \leq 2d)$, so that, for each reference subsequence $x_{i,j}$, all l-mers $x_{i',j}$ $(i' = 1, 2, \ldots, t,\ i' \neq i; j = 1, 2, \ldots, n - l + 1)$ in the whole sequences, except the ith sequence X_i such that $d_H(x_{i,j}, x_{i',j}) \leq k$, are selected to construct a cluster subset; that is

$$C(x_{i,j}, X) = \{x_{i,j}\} \bigcup_{i'=1 \wedge i' \neq i}^{t} B(x_{i,j}, X_{i'}), \tag{1}$$

where $B(x_{i,j}, X_{i'}) = \{x_{i',j} : x_{i',j} \in_l X_{i'}, d_H(x_{i,j}, x_{i',j}) \leq k\}$ represents the set of the selected l-mers in the i'th sequence $X_{i'}$ and $x_{i',j} \in_l X_{i'}$ if and only if $x_{i',j}$ is an $l-$mer of the sequence $X_{i'}$.

To set a proper threshold k, two probabilistic expressions are employed. The first is the probability of the Hamming distance between two random l-mers x_1 and x_2 being no more than k [4]:

$$p_k = p(d_H(x_1, x_2) \leq k) = \sum_{i=0}^{k} \binom{l}{i} \left(\frac{3}{4}\right)^{i} \left(\frac{1}{4}\right)^{l-i}. \tag{2}$$

The other is the probability of the Hamming distance between two selected motif instances m_1 and m_2 being no more than k: [18].

$$p_{kmotif} = p(d_H(m_1, m_2) \leq k). \tag{3}$$

Now, we describe the method for calculating p_{kmotif}. Given two motif instances, m_1 and m_2, of the same motif m_0 with up to d mutations, the distances between m_1, m_2, and m_0 satisfy $d_H(m_0, m_1) = \alpha$ and $d_H(m_0, m_2) = \beta$, $(0 \leq \alpha \leq d, 0 \leq \beta \leq d)$. Thus, the probability p_{kmotif} can be calculated as

$$\begin{aligned} p_{kmotif} &= p(d_H(m_1, m_2) \leq k) \\ &= p(d_H(m_1, m_2) \leq k | d_H(m_0, m_1) = \alpha, d_H(m_0, m_2) = \beta) \times p(\alpha, \beta), \end{aligned} \tag{4}$$

where $p(d_H(m_1, m_2) \leq k | d_H(m_0, m_1) = \alpha, d_H(m_0, m_2) = \beta)$ represents the conditional probability of $p(d_H(m_1, m_2) \leq k)$ given $d_H(m_0, m_1) = \alpha$, $d_H(m_0, m_2) = \beta$, such that

$$\begin{aligned} &p(d_H(m_1, m_2) \leq k | d_H(m_0, m_1) = \alpha, d_H(m_0, m_2) = \beta) \\ &= \begin{cases} \sum\limits_{i=[\frac{\alpha+\beta-k}{2}]+1}^{min(\alpha,\beta)} \frac{\binom{\alpha}{i} \times \binom{l-\alpha}{\beta-i} \times 3^{\beta}}{\binom{l}{\beta} \times 3^{\beta}} & k < \alpha + \beta \leq 2d, \\ 1 & 0 < \alpha + \beta \leq k, \end{cases} \end{aligned} \tag{5}$$

and $p(\alpha, \beta)$ represents the probability of $d_H(m_0, m_1) = \alpha$ and $d_H(m_0, m_2) = \beta$; that is

$$p(\alpha, \beta) = p(d_H(m_0, m_1) = \alpha, d_H(m_0, m_2) = \beta). \tag{6}$$

As $d_H(m_0, m_1) = \alpha$ and $d_H(m_0, m_2) = \beta$ are independent, we have

$$p(\alpha, \beta) = p(d_H(m_0, m_1) = \alpha) \times (d_H(m_0, m_2) = \beta), \tag{7}$$

$$p(d_H(m_0, m_1) = \alpha) = \binom{d}{\alpha} \frac{3^\alpha}{4^d}, \tag{8}$$

$$p(d_H(m_0, m_2) = \alpha) = \binom{d}{\beta} \frac{3^\beta}{4^d}. \tag{9}$$

Combining Equations (5)–(9), we have

$$\begin{aligned} p_{kmotif} &= p(d_H(m_1, m_2) \leq k), \\ &= \begin{cases} \sum\limits_{0 \leq \alpha, \beta \leq d} \dfrac{\left(\sum\limits_{i=[\frac{\alpha+\beta-k}{2}]+1}^{min(\alpha,\beta)} \binom{\alpha}{i}\binom{l-\alpha}{\beta-i} \right) \times 3^\beta}{\binom{l}{\beta} \times 3^\beta} \times \binom{2d}{\alpha} \frac{3^\alpha}{4^{2d}} \times \binom{2d}{\beta} \frac{3^\beta}{4^{2d}} & k < \alpha + \beta \leq 2d, \\ \left(\sum\limits_{0 \leq \alpha, \beta \leq d} 1 \right) \times \binom{2d}{\alpha} \frac{3^\alpha}{4^{2d}} \times \binom{2d}{\beta} \frac{3^\beta}{4^{2d}} & 0 < \alpha + \beta \leq k. \end{cases} \end{aligned} \tag{10}$$

Having calculated the two probabilities p_k and p_{kmotif}, we now describe how to set the proper threshold k, in order to construct the cluster subsets which contain as few background l-mers as possible while including sufficient motif instances. A larger value of p_{kmotif} indicates more motif instances belong to the cluster subset; however, a smaller value of p_k suggests that fewer background l-mers appear in the same cluster subset. Therefore, the threshold k should be set in a way that ensures that the value of p_{kmotif} is large enough, compared to the value of p_k.

To demonstrate this issue, let us consider the (18, 5) problem instance as an example. The values of p_k and p_{kmotif} are shown in Table 1. When $k = 7$, the value of p_{kmotif} is 0.7414, which allows us to obtain sufficient motif instances, whereas the value of p_k is 0.0012 which, in turn, allows us to reduce the scale of background l-mers in the same cluster subset. Therefore, the optimal value of k is 7.

Table 1. Values of p_k and p_{kmotif} under different values of k for (18, 5) problem instance.

k	5	6	7	8	9	10
p_k	3.42×10^{-5}	2.31×10^{-4}	0.0012	0.0054	0.0193	0.0569
p_{kmotif}	0.2303	0.4741	0.7414	0.9242	0.9915	1

2.2. Filter Cluster Subsets

As is known, the true motif instances must exist in one of these $h \times (n - l + 1)$ cluster subsets $C(x_{i,j}, X)$, which are constructed with the reference subsequences $x_{i,j}$ $(i = 1, 2, \ldots, h;$ $j = 1, 2, \ldots, n - l + 1)$ from the first h sequences. However, with a great number of total cluster subsets, the identification of the cluster subsets that contain the true motif instances is highly time-consuming as most of these cluster subsets have redundant background l-mers.

To filter the interference cluster subsets, a threshold p_{occ}^f (i.e., an occurrence frequency) [29] is employed with the purpose of analyzing the probability of a random motif instance x' occurring in a given sequence.

$$p_{occ}^f = \sum_{i=0}^{d} \binom{d}{i} \times p_{mut}^i \times (1 - p_{mut}^i)^{d-i} \times \frac{1}{\binom{l}{i} \times 3^i}, \tag{11}$$

where p_{mut} is the probability of a character mutation in one position. For each sequence $X_{i'}(i' = 1, 2, \ldots, t, i' \neq i)$, if the number of the selected l-mers in $B(x_{i,j}, X_{i'})$, denoted as $|B(x_{i,j}, X_{i'})|$, is greater than or equal to $p^f_{occ} \times (n - l + 1)$,

$$|B(x_{i,j}, X_{i'})| \geq p^f_{occ} \times (n - l + 1), \tag{12}$$

then the sequence $X_{i'}$ may contain a motif instance and is stored in a set $N(i, j)$. For each $N(i, j)$, if the number of the selected sequences in $N(i, j)$, denoted as $|N(i, j)|$, is not less than q,

$$|N(i, j)| \geq q, \tag{13}$$

then there are at least q possible motif instances in the corresponding cluster subset $C(x_{i,j}, X)$, the reference subsequence $x_{i,j}$ is considered as a potential motif instance, and the corresponding cluster subset $C(x_{i,j}, X)$ is a candidate cluster subset $C_{candidate}(x_{i,j}, X)$.

2.3. Refine Cluster Subsets

Due to the occurrence frequency p^f_{occ} and the relatively small Hamming distance k between the reference subsequence and the selected l-mers, only a limited number of cluster subsets, which contains a small number of the selected l-mers, are retained as candidate cluster subsets for further Affinity Propagation (AP) clustering. In order to quickly produce highly conserved cluster subsets, we apply AP clustering to each candidate cluster subset. Compared to other clustering approaches, AP clustering can cluster large-scale data sets efficiently by exchanging messages between data points.

2.3.1. Affinity Propagation (AP) Clustering

As demonstrated in our recent work [32], it is possible to speed up AP clustering and improve its accuracy with the adapted similarity $s(i, k)$, which is based on pair-wise constraints and a variable-similarity measure [33]. The adapted similarity, $s(i, k)$, between two l-mers $x_{i,j}$ and $x_{k,j}$ is defined as

$$s(i, k) = -\rho \times d_H(x_{i,j}, x_{k,j}) \times L(x_{i,j}, x_{k,j}, X), \tag{14}$$

where

$$\rho = \begin{cases} R_1 & \text{if } d_H(x_{i,j}, x_{k,j}) \in (0, k] \\ R_2 & \text{if } d_H(x_{i,j}, x_{k,j}) \in (k, 2k] \\ +\infty & \text{if } d_H(x_{i,j}, x_{k,j}) \in (2k, 4k] \end{cases} \tag{15}$$

$$L(x_{i,j}, x_{k,j}, X) = \begin{cases} +\infty & \text{if } x_{i,j} \in_l X_p, x_{k,j} \in_l X_q, p = q \\ 1 & \text{otherwise} \end{cases}. \tag{16}$$

Note that $R_1 \in (1, +\infty)$ and $R_2 \in (0, 1]$.

For each candidate cluster subset, based on the adapted similarity $s(i, k)$, AP clustering recursively calculates two types of messages. The first type is the responsibility $r(i, k)$, which reflects the suitability of point $x_{k,j}$ as the exemplar for point $x_{i,j}$. The other type is the availability $a(i, k)$, which indicates how suitable it would be for a point $x_{i,j}$ to choose the point $x_{k,j}$ as its exemplar:

$$r(i, k) = s(i, k) - \max_{x_{k',j} \neq x_{k,j}} \{a(i, k') + s(i, k')\}, \tag{17}$$

$$a(i,k) = min\{0, r(k,k) + \sum_{x_{i',j} \neq \{x_{i,j}, x_{k,j}\}} max\{0, r(i',k)\}\} \text{ if } x_{i,j} \neq x_{k,j}, \tag{18}$$

$$a(k,k) = \sum_{x_{i',j} \neq x_{k,j}} max\{0, r(i',k)\}\}. \tag{19}$$

When the AP clustering converges, a set of l-mers in the produced cluster subset are selected as exemplars $e(i)$ associated to the point $x_{i,j}$:

$$e(i) = \arg\max_{x_{k,j}} \{r(i,k) + a(i,k)\}. \tag{20}$$

2.3.2. Cluster Subset Refinement

To select an adequate number of desired cluster subsets with more motif instances but less background l-mers, we set an interval [min size, max size] = $[t-q, t]$ to further refine the cluster subsets $C_{AP}(x_{i,j}, X)$, which is produced, by AP clustering, using a reference l-mer $x_{i,j}$. Regarding the number of the l-mers in each cluster subset $C_{AP}(x_{i,j}, X)$, we conclude that there are three cases:

(i) In the case where the number is less than $t-q$ (for example, $|C_{AP}(x_{i,j}, X)| < t-q$), such a small number of l-mers is not enough to create a cluster subset $C_{AP}(x_{i,j}, X)$ which represents a true motif, so we consider it as an invalid cluster subset.

(ii) In the case where the number is between $t-q$ and t (for example $t-q < |C_{AP}(x_{i,j}, X)| \leq t$), we consider it as a valid cluster subset.

(iii) In the case where the number is more than t (for example $|C_{AP}(x_{i,j}, X)| > t$), the size of the cluster subset is so large that it may include too many background l-mers and, so, we use a greedy strategy to select t l-mers from $C_{AP}(x_{i,j}, X)$ to form $C_{valid}(x_{i,j}, X)$. First, $C_{valid}(x_{i,j}, X)$ is initialized with the AP clustering exemplar $e(i)$. Then, an l-mer $x_{r,j}$ from $C_{AP}(x_{i,j}, X) - C_{valid}(x_{i,j}, X)$ is repeatedly chosen, following Equations (21) and (22), and added to $C_{valid}(x_{i,j}, X)$ until $|C_{valid}(x_{i,j}, X)| = t$):

$$x_{r,j} = \arg\max_{x_{i,j} \in C_{AP} - C_{valid}} \sum_{x_{k,j} \in C_{valid}} sim(x_{i,j}, x_{k,j}), \tag{21}$$

$$sim(x_{i,j}, x_{k,j}) = \frac{len(x_{i,j}, x_{k,j})}{|x_{i,j}| + |x_{k,j}| - len(x_{i,j}, x_{k,j})}, \tag{22}$$

where $len(x_{i,j}, x_{k,j})$ is the length of the maximum intersection of $x_{i,j}$ and $x_{k,j}$.

To appropriately sort the valid cluster subsets $C_{valid}(x_{i,j}, X)$, we use the information content (IC) [34], and the cluster subset with the maximum value of IC is considered as the true motif model:

$$IC(C_{valid}(x_{i,j}, X)) = \sum_{m=1}^{l} \sum_{w=1}^{4} p_{w,m} \log \frac{p_{w,m}}{p_{w,0}}, \tag{23}$$

where $p_{w,m}$ is the probability of the character $w \in A, T, C, G$ at the position m of the l-mer $x_{i,j}$, and $p_{w,0}$ is the corresponding background probability.

Based on the above described three steps, the whole AP-ChIP Algorithm 1 is described as follows:

```
Algorithm 1: AP-ChIP algorithm
   Input: $l, d, q, X = \{X_1, X_2, \ldots, X_t\}$
   Output: $(l, d)$ motif $X_{motif}$
 1 $h \leftarrow t - q + 1$;
 2 for $i \leftarrow 1$ to $h$ do
 3     for each l-mer $x_{i,j} \in X_i$ do
 4         $C(x_{i,j}, X) \leftarrow \emptyset, N(i,j) \leftarrow \emptyset$;
 5         for $i' \leftarrow 1$ to $t$, $i' \neq i$ do
 6             $B(x_{i,j}, X_{i'}) \leftarrow \emptyset$;
 7             for each l-mer $x_{i',j} \in X_{i'}$ do
 8                 if $d_H(x_{i,j}, x_{i',j}) \leq k$ then
 9                     $B(x_{i,j}, X_{i'}) \leftarrow B(x_{i,j}, X_{i'}) \cup \{x_{i',j}\}$;
10                 end
11             end
12             $C(x_{i,j}, X) \leftarrow C(x_{i,j}, X) \cup B(x_{i,j}, X_{i'})$;
13             if $|B(x_{i,j}, X_{i'})| \geq p^f_{occ} \times (n - l + 1)$ then
14                 $N(i,j) \leftarrow N(i,j) \cup \{X_{i'}\}$
15             end
16         end
17     end
18     if $|N(i,j)| \geq q$ then
19         $C_{candidate}(x_{i,j}, X) \leftarrow C(x_{i,j}, X)$
20     end
21 end
22 for each $C_{candidate}(x_{i,j}, X)$ do
23     Use AP clustering and a greedy strategy to generate valid cluster subsets $C_{valid}(x_{i,j}, X)$;
24 end
25 $IC_{max} \leftarrow 0$;
26 for each $C_{valid}(x_{i,j}, X)$ do
27     if $IC(C_{valid}(x_{i,j}, X)) > IC_{max}$ then
28         $IC_{max} \leftarrow IC(C_{valid}(x_{i,j}, X))$;
29     end
30 end
31 get $X_{motif}$ from $IC_{max}$
```

Steps 2–17 describe the process of constructing the cluster subsets. Steps 18–21 describe the filtration of the cluster subsets. Steps 22–31 describe the refinement of the cluster subsets and the verification of the motif with the maximum IC score.

3. Results

3.1. Results on Simulated Data

Simulated data provide quantitative measures to test the performance of the AP-ChIP Algorithm 1. As in [29], we generate the simulated data as follows:

First, we generated t independent and identically distributed (i.i.d) sequences of length n and a motif m of length l. Second, we randomly generated q $(0 < q \leq t)$ motif instances, each of which differed from the motif m at up to d positions. Third, the q motif instances were placed in a random position in a random selection of q sequences selected out of the t sequences. We, then, implemented the AP-ChIP Algorithm 1, using Matlab on a computer with a 2.6 GHZ processor and 4 Gbyte memory. The final experimental results consisted of the averages of five trials of simulated data experiments.

To evaluate the motif prediction accuracy, the nucleotide level performance coefficient (nPC), as defined by Pevezner and Sze [2], was used:

$$nPC = \frac{|K \cap P|}{|K \cup P|}, \tag{24}$$

where K is the set of nucleotide positions in the true motif and P is the corresponding set of nucleotide positions in the predicted motif. The value of nPC is between 0 and 1; the larger the value of nPC, the higher the accuracy of the predicted motif. We used the $2d$ neighborhood probability p_{2d} [8] to select a group of (l, d) motif instances. The larger the value of p_{2d}, the weaker the corresponding (l, d) problem instance becomes:

$$p_{2d} = P(d_H(x_1, x_2) \leq 2d) = \sum_{i=0}^{2d} \binom{l}{i} \left(\frac{3}{4}\right)^i \left(\frac{1}{4}\right)^{l-i}. \tag{25}$$

In what follows, according to different values of $\alpha = \frac{q}{t}$ (i.e., the ratio of the sequences containing motif instances to all the sequences), we designed two groups of experiments, both of which consisted of two sub-experiments, to test the performance of the AP-ChIP Algorithm 1 on simulated data sets.

In the first group of experiments, we set $\alpha = 100\%$ for both sub-experiments. We compared the performance of the AP-ChIP Algorithm 1 with that of the widely used motif finding algorithms MEME [13], VINE [14], and Projection [16].

In the first sub-experiment, we set the number of sequences as $t = 20$ with sequence length $n = 600$. The running time and the predicted accuracy of these algorithms are shown in Table 2. For the instances (12, 2) and (15, 3) with $p_{2d} < 0.05$, the AP-ChIP Algorithm 1 achieved near-optimal predicted accuracy within a relatively short time. For the instances (15, 4), (14, 4), (25, 8), and (21, 7) with $p_{2d} \geq 0.05$, the predicted accuracy of the AP-ChIP Algorithm 1 was over 90% and the running time of the AP-ChIP Algorithm 1 remained competitive, compared to the other algorithms.

Table 2. Comparisons on (l, d) problem instances with $t = 20$, $n = 600$, and $\alpha = 100\%$.

(l, d)	p_{2d}	MEME	VINE	Projection	AP-ChIP
(12, 2)	0.0028	0.68 (4 s)	1.00 (8 s)	0.86 (10 s)	0.98 (18 s)
(15, 3)	0.0042	0.73 (7 s)	1.00 (9 s)	0.82 (1.3 m)	1.00 (23 s)
(15, 4)	0.0566	0.87 (8 s)	0.96 (5.6 m)	0.89 (4.2 m)	0.97 (36 s)
(14, 4)	0.1117	0.84 (10 s)	0.95 (8.3 m)	0.80 (27.4 m)	0.96 (47 s)
(25, 8)	0.1494	0.91 (12 s)	0.93 (9.8 m)	0.78 (32.6 m)	0.94 (1.1 m)
(21, 7)	0.2564	0.87 (28 s)	0.92 (11.2 m)	0.76 (48.7 m)	0.91 (58 s)

In general, it is easy to find the true motif by increasing the sequence number and decreasing its length. Therefore, in the second sub-experiment, we set the sequence number as $t = 1000$ with sequence length $n = 200$. Table 3 shows that, for all (l, d) problem instances with $p_{2d} \geq 0.05$, the predicted accuracy of the AP-ChIP Algorithm 1 is over 90% with the computational costs being satisfactory.

Next, in the second group of experiments, to simulate a real ChIP-Seq data set, we set $\alpha = 90\%$ for both sub-experiments. This is because in real ChIP-Seq data set, most but not all of the sequences contain motif instances.

Table 3. Comparisons on (l, d) problem instances with $t = 1000$, $n = 200$, $\alpha = 100\%$.

(l, d)	p_{2d}	MEME	VINE	Projection	AP-ChIP
(12, 3)	0.0540	0.94 (8 s)	0.96 (2.4 m)	0.91 (3.1 m)	0.97 (21 s)
(11, 3)	0.1146	0.86 (10 s)	0.95 (5.2 m)	0.84 (4.3 m)	0.96 (33 s)
(13, 4)	0.2060	0.83 (10 s)	0.93 (8.1 m)	0.78 (36.4 m)	0.95 (36 s)
(15, 5)	0.3135	0.78 (11 s)	0.84 (9.6 m)	0.74 (46.7 m)	0.93 (34 s)
(17, 6)	0.4261	0.70 (13 s)	0.83 (18.6 m)	0.72 (53.6 m)	0.92 (38 s)
(19, 7)	0.5346	0.68 (17 s)	0.75 (24.5 m)	0.70 (1.2 h)	0.90 (40 s)

In the first sub-experiment, we test the validity of the AP-ChIP Algorithm 1 on the simulated ChIP-Seq data set for the identification of (l, d) motifs with $t = 1000$ and $n = 200$. We choose $p_{2d} = 0.05$ to select a group of (l, d) problem instances. The reason for this choice is that $p_{2d} = 0.05$ is approximately the same as the p_{2d} value of the (15, 4) problem instance, which is one of the most popular benchmarks for (l, d) problem instance. The running time and the predicted motif by the AP-ChIP Algorithm 1 are shown in Table 4. For each (l, d) problem instance, the AP-ChIP Algorithm 1 finds almost the same motif as the published one and also runs quite efficiently.

Table 4. The results on (l, d) problem instances with $p_{2d} = 0.05$, $\alpha = 90\%$.

(l, d)	Time	Predicted Motif	Published Motif
(9, 2)	43 s	TTATCCCTC	TTATCCCTC
(12, 3)	34 s	TTTCCCGTCTGC	CTTTCCCGTCTG
(15, 4)	42 s	GGTTGRAGCTTAGGG	GGTTGGAGCTTAGGG
(18, 5)	38 s	CTTTGCCATATCCATAGG	TTTGCCATATCCATAGGC
(21, 6)	36 s	CAGGTAAACCATATTAAATTA	AGGTAAACCATATTAAATTAC

R: A,G.

In the second sub-experiment, in order to further demonstrate the performance of the AP-ChIP Algorithm 1 on the simulated ChIP-Seq data set, we compared the AP-ChIP Algorithm 1 against some established motif-finding algorithms in the following two aspects: (i) Different values of p_{2d}, ranging from 0.05 to 0.5, with fixed $\alpha = 90\%$, and (ii) different values of α floating from 0.7 to 1 with fixed $(l, d) = (9, 2)$. Although a genome-wide ChIP-Seq data set typically has thousands to tens of thousands of sequences, using 20% to 50% of the ChIP-Seq data set is usually adequate for obtaining a good estimate of the true motifs. MEME-ChIP, a well-known algorithm for discovering motifs in ChIP-Seq data sets, is able to well identify (l, d) motifs with only 600 sequences. Thus, it was reasonable to set the sequence number as $t = 600$ and sequence length as $n = 200$ for motif discovery in our experiments.

First, we compared the running time and prediction accuracy of the AP-ChIP Algorithm 1 with those of the three compared algorithms, MEME-ChIP [20], ChIP-Munk [24], and FMotif [25], on different values of p_{2d} and with fixed $\alpha = 90\%$. As shown in Table 5, for each (l, d) problem instance, the AP-ChIP Algorithm 1 could solve it in a relatively short time, and its prediction accuracy was better than those of the three compared algorithms. Specifically, with increasing values of l and d, FMotif found it difficult to find exact motifs.

Table 5. Comparison of (l, d) problem instances with $t = 600$, $n = 200$, and $\alpha = 90\%$.

(l, d)	p_{2d}	MEME-ChIP	ChIP-Munk	FMotif	AP-ChIP
(9, 2)	0.049	0.96 (12 s)	0.96 (1.8 m)	1.00 (47 s)	1.00 (43 s)
(11, 3)	0.114	0.94 (24 s)	0.92 (2.0 m)	0.99 (7.9 m)	0.98 (46 s)
(13, 4)	0.205	0.90 (38 s)	0.83 (2.4 m)	0.98 (1.45 h)	0.93 (58 s)
(15, 5)	0.319	0.85 (42 s)	0.80 (8.2 m)	–	0.92 (1.1 m)
(17, 6)	0.426	0.80 (45 s)	0.78 (9.6 m)	–	0.89 (1.3 m)
(19, 7)	0.534	0.78 (48 s)	0.76 (10.7 m)	–	0.87 (1.6 m)

Next, we compared the prediction accuracy of AP-ChIP Algorithm 1 with those of the algorithms MEME [13], MEME-ChIP [20], DREME [22], and FMotif [25], on different values of α floating from 0.7 to 1 and fixed $(l, d) = (9, 2)$. As the ratio of a motif instance $\alpha = \frac{q}{t}$ has a strong effect on the prediction accuracy, it was necessary for us to test the prediction accuracy of AP-ChIP Algorithm 1 on different values of α. It is rather cumbersome to identify the true motif when the value of α is small. FMotif is a powerful, exhaustive algorithm for finding exact short (l, d) motifs $(l \leq 10, d \leq 2)$ contained in ChIP-Seq data sets. As shown in Figure 1, the prediction accuracy of AP-ChIP Algorithm 1 was nearly the same as that of FMotif, and was higher than that of MEME-ChIP, MEME, and DREME.

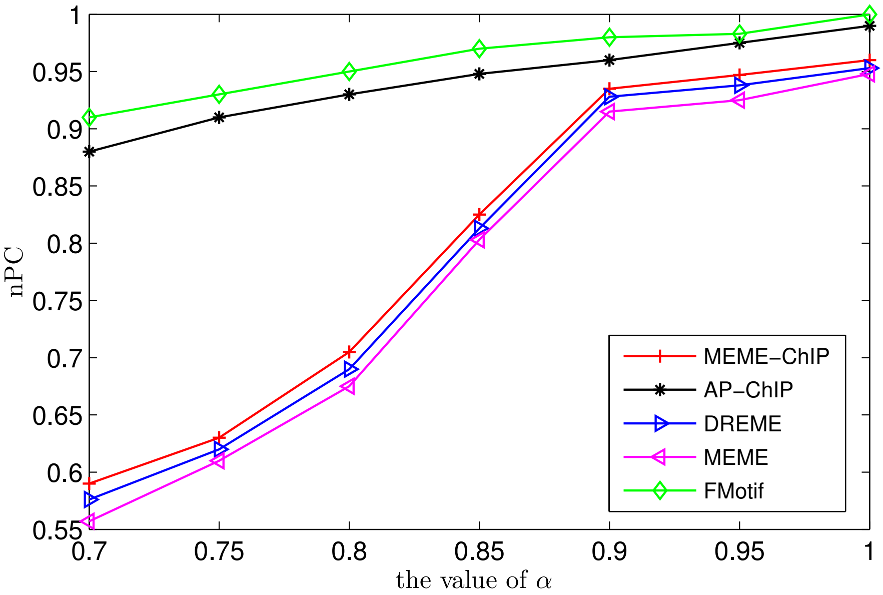


Figure 1. Prediction accuracy for different values of α.

3.2. Results on Real Data

First, we tested the validity of the AP-ChIP Algorithm 1 for the identification of real motifs using a diverse set of real ChIP-Seq data sets; specifically, on 12 differently sized mESC data sets (Nanog, Oct4, Sox2, Esrrb, Zfx, Klf4, c-Myc, n-Myc, Tcfcp21l, Smad1, STAT3, and CTCF) [35]. We compared the motifs detected by the AP-ChIP Algorithm 1 with the motifs published by Chen et al. [35], and presented them in the form of sequence logos [36], which graphically represent the degree of motif conservation, as measured by relative entropy. Table 6 shows the running times and the predicted motifs of the AP-ChIP Algorithm 1. For each data set, the AP-ChIP Algorithm 1 was capable of finding motifs highly similar to the published ones within a reasonable time.

Moreover, to better show the results, we compared AP-ChIP 1 with MEME-ChIP on nine differently sized ENCODE data sets (Nfyb, Hnf4, Elf1, Ets, Egr1, Yy1, Six5, Srf, and Tal1) [37], where the TFBSs were referenced in the JASPAR database [38]. Table 7 shows the published motifs and the motifs predicted by the two algorithms. The motifs are also shown in the form of sequence logos. AP-ChIP 1 could successfully find a motif similar to the published motif for each data set, while, for some data sets MEME-ChIP failed to accurately predict the motif (e.g., in the Elf1 data set), or lost information on individual bases (e.g., in the Tal dataset).

Table 6. Results on the mESC data set.

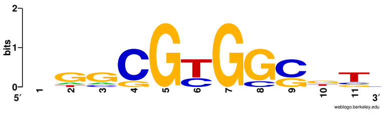
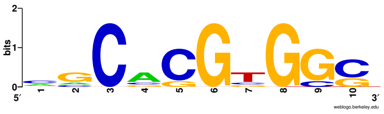
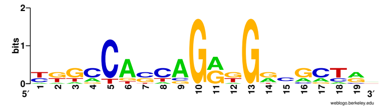
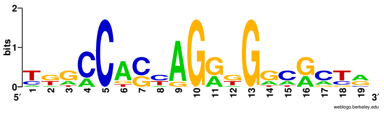
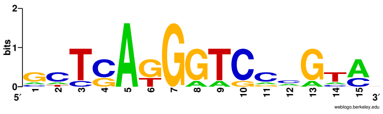
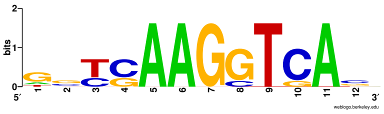
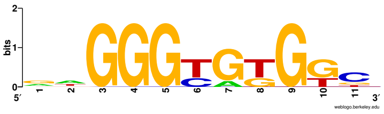
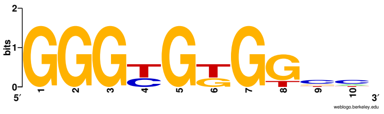
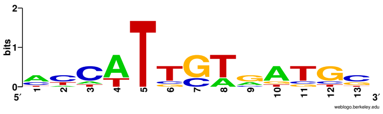
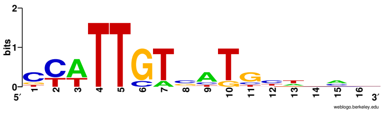
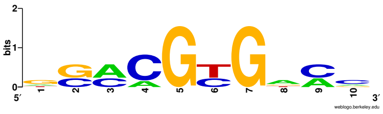
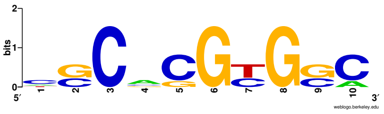
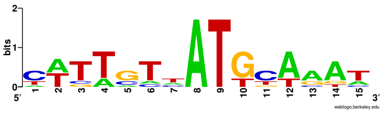
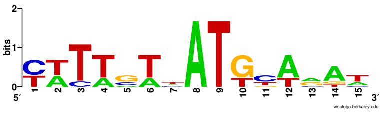
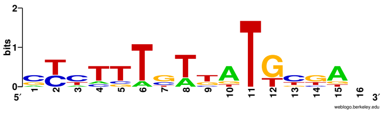
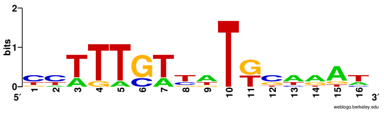
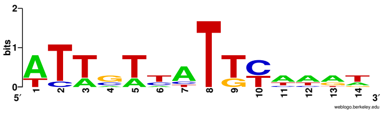
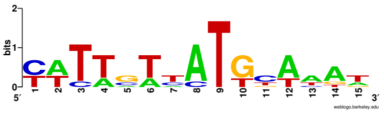
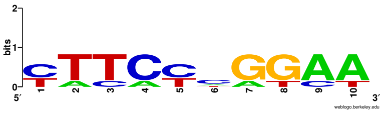
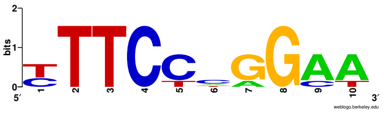
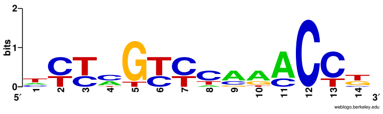
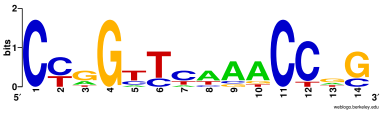
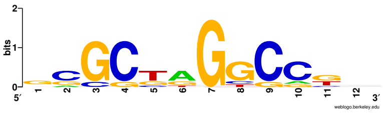
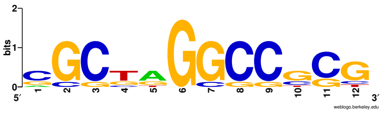
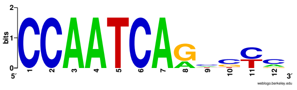
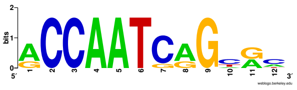
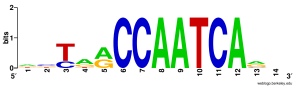
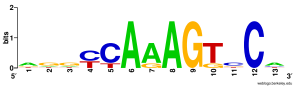
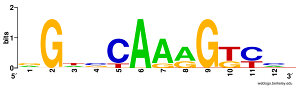
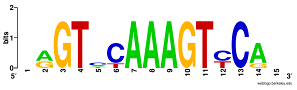
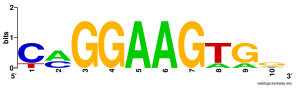
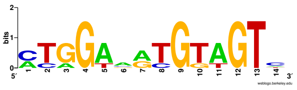
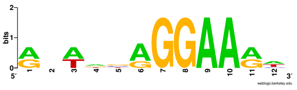
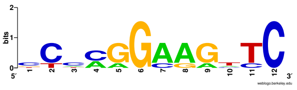
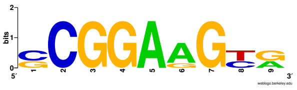
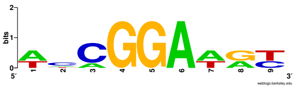
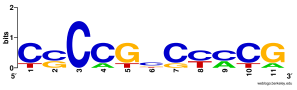
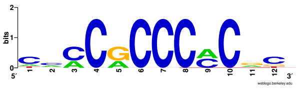
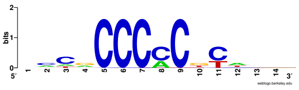
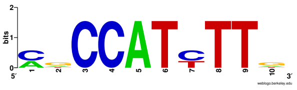
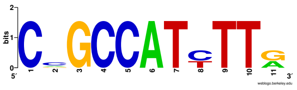
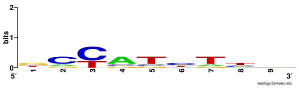
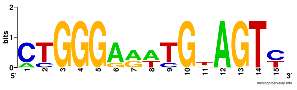
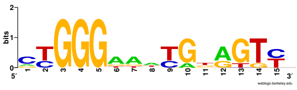
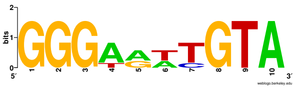
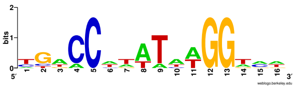
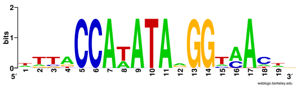
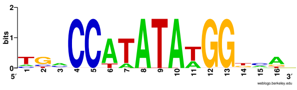
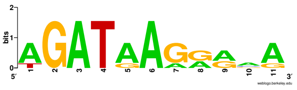
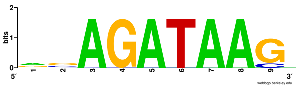
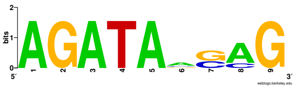
Data Set (Seq #)	Time	Predicted Motif	Published Motif
c-Myc (3422)	125 s		
CTCF (39609)	19 s		
Esrrb (21647)	10 s		
Klf4 (10875)	138 s		
Nanog (10343)	12 s		
n-Myc (7182)	36 s		
Oct4 (3761)	48 s		
Smad1 (1126)	12 s		
Sox2 (4525)	15 s		
STAT3 (2546)	27 s		
Tcfcp211 (26910)	11 s		
Zfx (10338)	68 s		

Table 7. Results on the ENCODE dataset.

Data Set (Seq #)	AP-ChIP Predicted Motif	MEME-ChIP Predicted Motif	Published Motif
Nfyb (10096)			
Hnf4 (11045)			
Elf1 (8611)			
Ets (5525)			
Egr1 (15400)			
Yy1 (2077)			
Six5 (4664)			
Srf (4903)			
Tal1 (25507)			

4. Concluding Remarks

In this paper, our goal was to find a method providing balance between time performance and prediction accuracy for TFBS discovery in ChIP-Seq data sets. Consequently, we aimed to obtain these results with high prediction accuracy in a relatively short time. To do so, we proposed a novel clustering-based algorithm named AP-ChIP 1. Firstly, to achieve high prediction accuracy, we set a threshold k to restrict the number of the selected l-mers in the candidate cluster subsets. Next, to obtain good time performance, we set the threshold p_{occ}^{f} in terms of probabilistic analysis, in order to filter the interferential candidate cluster subsets. Furthermore, a powerful data clustering method, AP clustering, was used to obtain the almost accurate motifs. Experimental results on both simulated and real ChIP-Seq datasets showed that the AP-ChIP Algorithm 1 not only discovers the motifs as consistently as the published ones, but also does so quite efficiently. This demonstrates that the AP-ChIP Algorithm 1 is a powerful new approach for ChIP-Seq data set analysis which provides a good trade-off between time performance and prediction accuracy.

Author Contributions: C.-x.S. and Y.Y. contribute for supervision, project administration, and formal analysis. C.-x.S., Y.Y., H.W., and W.-h.W. contribute for methodology and writing original draft preparation. The final draft we written by C.-x.S. and H.W.

Funding: This work is supported by the National Natural Science Foundation of China (grant nos. 61702291, 61772102, 61472058,11531001), China Postdoctoral Science Foundation (grant no. 2018M632095), Program for Science & Technology Innovation Talents in Universities of Henan Province (grant no. 19HASTIT029), the Key Research Project in Universities of Henan Province (grant nos. 19B110011, 19B630015), the Simons Foundation (grant no. 245307), and the Scientific Research Starting Foundation for High-level Talents of Pingdingshan University (grant no. PXY-BSQD2017006).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Stormo, G.D. DNA binding sites: Representation and discovery. *Bioinformatics* **2000**, *16*, 16–23. [CrossRef] [PubMed]
2. Pevzner, P.A.; Sze, S.H. Combinatorial approaches to finding subtle signals in DNA sequences. In *ISMB*; American Association for Artificial Intelligence: Menlo Park, CA, USA, 2000; Volume 8, pp. 269–278.
3. Bailey, T.; Elkan, C. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **1994**, *2*, 28–36.
4. Yu, Q.; Huo, H.; Zhang, Y.; Guo, H. PairMotif: A new pattern-driven algorithm for planted (l, d) DNA motif search. *PLoS ONE* **2012**, *7*, e48442. [CrossRef] [PubMed]
5. Chin, F.Y.; Leung, H.C. Voting algorithms for discovering long motifs. In Proceedings of the 3rd Asia-Pacific Bioinformatics Conference, Singapore, 17–21 January 2005; pp. 261–271.
6. Davila, J.; Balla, S.; Rajasekaran, S. Fast and practical algorithms for planted (l, d) motif search. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2007**, *4*, 544–552. [CrossRef] [PubMed]
7. Dinh, H.; Rajasekaran, S.; Kundeti, V.K. PMS5: An efficient exact algorithm for the (l, d)-motif finding problem. *BMC Bioinform.* **2011**, *12*, 410. [CrossRef] [PubMed]
8. Ho, E.S.; Jakubowski, C.D.; Gunderson, S.I. iTriplet, a rule-based nucleic acid sequence motif finder. *Algorithms Mol. Biol.* **2009**, *4*, 14. [CrossRef] [PubMed]
9. Dinh, H.; Rajasekaran, S.; Davila, J. qPMS7: A fast algorithm for finding (l, d)-motifs in DNA and protein sequences. *PLoS ONE* **2012**, *7*, e41425. [CrossRef] [PubMed]
10. Nicolae, M.; Rajasekaran, S. Efficient sequential and parallel algorithms for planted motif search. *BMC Bioinform.* **2014**, *15*, 34. [CrossRef] [PubMed]
11. Schneider, T.D. Consensus Sequence Zen. *Appl. Bioinform.* **2002**, *1*, 111–119.
12. Quang, D.; Xie, X. EXTREME: An online EM algorithm for motif discovery. *Bioinformatics* **2014**, *30*, 1667–1673. [CrossRef] [PubMed]
13. Bailey, T.L.; Williams, N.; Misleh, C.; Li, W.W. MEME: Discovering and analyzing DNA and protein sequence motifs. *Nucleic Acids Res.* **2006**, *34*, W369–W373. [CrossRef] [PubMed]
14. Huang, C.W.; Lee, W.S.; Hsieh, S.Y. An improved heuristic algorithm for finding motif signals in DNA sequences. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2010**, *8*, 959–975. [CrossRef] [PubMed]
15. Lawrence, C.E.; Altschul, S.F.; Boguski, M.S.; Liu, J.S.; Neuwald, A.F.; Wootton, J.C. Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment. *Science* **1993**, *262*, 208–214. [CrossRef] [PubMed]
16. Buhler, J.; Tompa, M. Finding motifs using random projections. *J. Comput. Biol.* **2002**, *9*, 225–242. [CrossRef] [PubMed]
17. Lee, N.K.; Li, X.; Wang, D. A comprehensive survey on genetic algorithms for DNA motif prediction. *Inf. Sci.* **2018**, *466*, 25–43. [CrossRef]
18. Wong, K.C. MotifHyades: Expectation maximization for de novo DNA motif pair discovery on paired sequences. *Bioinformatics* **2017**, *33*, 3028–3035. [CrossRef]
19. Thompson, J.D.; Higgins, D.G.; Gibson, T.J. CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* **1994**, *22*, 4673–4680. [CrossRef] [PubMed]
20. Machanick, P.; Bailey, T.L. MEME-ChIP: Motif analysis of large DNA datasets. *Bioinformatics* **2011**, *27*, 1696–1697. [CrossRef]
21. Hu, M.; Yu, J.; Taylor, J.M.; Chinnaiyan, A.M.; Qin, Z.S. On the detection and refinement of transcription factor binding sites using ChIP-Seq data. *Nucleic Acids Res.* **2010**, *38*, 2154–2167. [CrossRef]
22. Bailey, T.L. DREME: Motif discovery in transcription factor ChIP-seq data. *Bioinformatics* **2011**, *27*, 1653–1659. [CrossRef]
23. Reid, J.E.; Wernisch, L. STEME: Efficient EM to find motifs in large data sets. *Nucleic Acids Res.* **2011**, *39*, e126. [CrossRef] [PubMed]
24. Kulakovskiy, I.; Boeva, V.; Favorov, A.; Makeev, V. Deep and wide digging for binding motifs in ChIP-Seq data. *Bioinformatics* **2010**, *26*, 2622–2623. [CrossRef] [PubMed]

25. Jia, C.; Carson, M.B.; Wang, Y.; Lin, Y.; Lu, H. A new exhaustive method and strategy for finding motifs in ChIP-enriched regions. *PLoS ONE* **2014**, *9*, e86044. [CrossRef] [PubMed]
26. Zambelli, F.; Pavesi, G. A faster algorithm for motif finding in sequences from ChIP-Seq data. In *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 201–212.
27. Yu, Q.; Wei, D.; Huo, H. SamSelect: A sample sequence selection algorithm for quorum planted motif search on large DNA datasets. *BMC Bioinform.* **2018**, *19*, 228. [CrossRef] [PubMed]
28. Sharov, A.A.; Ko, M.S. Exhaustive search for over-represented DNA sequence motifs with CisFinder. *DNA Res.* **2009**, *16*, 261–273. [CrossRef] [PubMed]
29. Yu, Q.; Huo, H.; Chen, X.; Guo, H.; Vitter, J.S.; Huan, J. An efficient motif finding algorithm for large DNA data sets. In Proceedings of the 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Belfast, UK, 2–5 November 2014; pp. 397–402.
30. Frey, B.J.; Dueck, D. Clustering by passing messages between data points. *Science* **2007**, *315*, 972–976. [CrossRef]
31. Yu, Q.; Huo, H.; Zhao, R.; Feng, D.; Vitter, J.S.; Huan, J. Reference sequence selection for motif searches. In Proceedings of the 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Washington, DC, USA, 9–12 November 2015; pp. 569–574.
32. Sun, C.; Huo, H.; Yu, Q.; Guo, H.; Sun, Z. An affinity propagation-based DNA motif discovery algorithm. *BioMed Res. Int.* **2015**, *2015*, 853461. [CrossRef]
33. Leone, M.; Weigt, M. Clustering by soft-constraint affinity propagation: Applications to gene-expression data. *Bioinformatics* **2007**, *23*, 2708–2715. [CrossRef]
34. Wang, D.; Lee, N.K. Computational discovery of motifs using hierarchical clustering techniques. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 1073–1078.
35. Chen, X.; Xu, H.; Yuan, P.; Fang, F.; Huss, M.; Vega, V.B.; Wong, E.; Orlov, Y.L.; Zhang, W.; Jiang, J.; et al. Integration of external signaling pathways with the core transcriptional network in embryonic stem cells. *Cell* **2008**, *133*, 1106–1117. [CrossRef]
36. Crooks, G.E.; Hon, G.; Chandonia, J.M.; Brenner, S.E. WebLogo: A sequence logo generator. *Genome Res.* **2004**, *14*, 1188–1190. [CrossRef]
37. Qu, H.; Fang, X. A Brief Review on the Human Encyclopedia of DNA Elements (ENCODE) Project. *Genom. Proteom. Bioinform.* **2013**, *11*, 135–141. [CrossRef] [PubMed]
38. Khan, A.; Fornes, O.; Stigliani, A.; Gheorghe, M.; Castro-Mondragon, J.A.; Van, d.L.R.; Bessy, A.; ChèNeby, J.; Kulkarni, S.R.; Tan, G. JASPAR 2018: Update of the open-access database of transcription factor binding profiles and its web framework. *Nucleic Acids Res.* **2017**, *77*, e43.