# Fast and accurate genome-scale identification of DNA-binding sites

David Martin, Vincent Maillol, Eric Rivals

# Fast and accurate genome-scale identification of DNA-binding sites

David Martin[1], Vincent Maillol[1], Eric Rivals[1,2*]
1. Laboratory of Informatics, Robotics and Microelectronics (LIRMM)
CNRS & Univ. Montpellier, Montpellier, France
2. Institut de Biologie Computationnelle (IBC), Montpellier, France,
*Email*: rivals@lirmm.fr

December 31, 2018

**Abstract**

**Motivation:** Discovering DNA binding sites in genome sequences is crucial for understanding genomic regulation. Currently available computational tools for finding binding sites with Position Weight Matrices of known motifs are often used in restricted genomic regions because of their long run times. The ever-increasing number of complete genome sequences points to the need for new generations of algorithms capable of processing large amounts of data.
**Results:** Here we present MOTIF, a new algorithm for seeking transcription factor binding sites in whole genome sequences in a few seconds. We propose a web service that enables the users to search for their own matrix or for multiple JASPAR matrices. Beyond its efficacy, the service properly handles undetermined positions within the genome sequence and provides an adequate output listing for each position the matching word and its score.
**Availability:** MOTIF is freely available for use through an interface at http://www.atgc-montpellier.fr/motif. The source code of the stand-alone search method of MOTIF is freely available at https://gite.lirmm.fr/rivals/motif.git. It is written in C++ and tested on Linux platforms.
**Contact:** motif@lirmm.fr

## 1 Introduction

Gene expression is a complex process requiring a tight regulation involving a wide range of mechanisms. Among them, regulation of transcription plays a key role in modulating the amount of mRNA produced by the RNA polymerase II. It involves a wide array of proteins, among which transcription factors [14] (TFs). These regulatory proteins typically bear a DNA-binding domain allowing them to interact with the DNA in the vicinity of the gene(s) they regulate, often found at the promoter and/or within more remote regulatory elements (enhancers) [9]. They also contain regulatory domains able to activate or repress the triggering of the polymerase, and eventually protein-protein interaction domains to recruit other actors of the transcriptional regulation machinery [8]. TFs bind to DNA in a sequence-specific manner: they are able to recognize a restricted set of nucleic acid segments called binding sites (TFBSs). Their binding specificity can be modeled computationally as the alignment of a collection of the TFBSs a given

*to whom correspondence should be addressed

TF recognizes. This alignment is then converted into a motif (or pattern) in the form of an alignment matrix, also referred to as position-frequency matrix (PFM) or position-count matrix (PCM) [15]. Such matrices are available through dedicated public (Jaspar [11], Hocomoco [6]) or private (TransFac [16]) databases, for motif search purpose. In this context, various transformations of a PFM into a position-weight matrix (PWM) have been proposed so far. The PWM is used to scan a DNA sequence and returns a score at each position. The higher the score, the better a sequence segment resembles to the matrix. Putative binding sites, or hits, are those DNA segments whose score is above a user-defined score threshold. The choice of this threshold is specific for each matrix and organism. Various methods exist to set the threshold, from the intuitive ones, like the percentage value of the matrix score range or the expected number of hits, to empirical methods like P-value calculations [13] or Receiver Operating Characteristic curves [1]. This last method requires that the TFBSs used to build the matrix are available in their genomic context, which is rarely the case in practice.

Recently, with the development of deep sequencing technologies, the number of completed genomic sequences has increased dramatically. Moreover, larger genomes can nowadays be sequenced such as plant genomes. Additionally, new technologies such as HT-Selex allow the characterization of DNA-binding profiles at a large scale [4]. This points the need for a new generation of algorithms capable of facing the analysis of this huge amount of sequence and pattern data. Here we present a new method, called MOTIF, for fast and accurate scanning of large genomes with PWMs. We also compare the implementation of our method with the fastest program known to date, MOODS [5], and show the gain in terms of speed, accuracy and ease of use.

# 2 Methods

## 2.1 Search algorithm

Contrarily to classic methods, MOTIF does not scan every possible position in the target genome. Instead, it uses an efficient strategy to generate all solutions (DNA words) from a given matrix passing the threshold score, and then maps them on the genome sequence using an optimized index structure (Figure 1). In some ways, our approach transforms PWM searching into a mapping problem. In practice, reference genomes are quite stable and users perform numerous searches on the same genome sequence. Scanning several times a genome from end to end for each search would be extremely long and redundant in terms of computation. Our approach takes advantage of this situation and adopts what is called an off-line strategy. In the first step, the genome sequence is preprocessed only once to build an index, which enables MOTIF to find the occurrences of any word in the genome. Indeed, the index offers a query procedure that takes a word as input and returns all positions of that word within the genome. This procedure takes an optimal time, proportional to the length of the word, rather than to the genome length with a scanning strategy. It can be used repeatedly to search for as many words as needed. However, querying the index one performs an exact search: it returns only the positions where the genome matches the word exactly. Let us now explain how index queries can be used in the context of search for PWMs.

Assume one searches for a PWM of width w with a given threshold score. Only words of length w can match that PWM. With 4 nucleotides, there exist $4^w$ possible words. However, only a subset of these can achieve a score larger than the threshold. Such a word is called a matching word. Instead of scanning the genome sequence to find whether each window of length w has a sufficient score (i.e., an on-line strategy), we adopt a combinatorial search. Our strategy is to list all possible words of length w whose score exceeds the threshold, and to output their occurrences

positions by querying the index. Typical width for transcription factor matrices ranges from 4 to 30 nucleotides/columns in public databases (Figure 2a, Supplementary Data), and brute-force enumeration for large widths would be excessively time-consuming. We developed an algorithm to perform this enumeration very quickly by cutting down branches of the solution tree through constrained programming. The underlying idea is to take advantage of the information content (IC) - i.e. the variability of letters that can be found in a column of the matrix (Figure 2a). The less variable a position, the higher its IC, and the stronger its contribution to the DNA segment score. Using this property, DNA strings can be generated exhaustively and efficiently by discarding those containing penalizing letters at strong IC positions, which would cause the segment score to fall below the threshold (Figure 1).

The $4^w$ possible words can be represented by a search tree with w levels, where each word appears in a leaf, and an internal node stores a prefix of the words in its subtree. Even for width below 30 nucleotides, testing all possible $4^w$ words and computing their score can be extremely long. We exploit an important property: since in a PWM model all positions are independent, the score of a word is the sum of the scores of each position. Hence, it does not depend on the order in which positions are summed. We use a combinatorial exploration of the tree that avoids visiting branches as soon as a partial score of an internal node is too low to allow any leaf below it to reach the threshold (Figure 1). This can be done by visiting nodes of the tree with Depth-First-Search strategy; it is a Branch and Bound algorithm. In the case of PWM search, we can improve on this for the following reason: the positions of a PWM do not contribute equally to the score. As explained above, their contribution depends on the IC of the corresponding matrix column. Hence, we order the positions according to the information content of the matrix columns (Figure 1). That way, the branch and bound algorithm first sets the positions that contribute the most to the score, and deeper in the tree those positions that contribute less. We implemented this strategy using constraint programming.

In practice, two cases arise:

1. The matrix width is small ($w \leq 10$). It is very fast to enumerate all possible DNA words (i.e. $4^w$ words) and compute their score. Only those with a score above the threshold are kept for the search.

2. The matrix width is large ($w > 10$).

The exhaustive enumeration option becomes too slow, for the number of words grows exponentially. Thus, we adopt a strategy based on constraint programming to limit the generation to only those words having a sufficient score. We take advantage of the score additivity (the total score sums the score for each column); this sum does not depend on the order in which the matrix columns are considered. The algorithm explores gradually and virtually the search tree in which all $4^w$ possible words are leaves (Figure 1). Our algorithm considers the columns one after the other in the order of decreasing information content, because the higher the information content of a column, the larger its contribution to the score. For a given column, it will consider all four possibilities one after the other by order of decreasing score. After visiting a node (column), the algorithm computes the partial score for the current set of columns, and deduces the amount of score missing to reach the threshold. If the partial scores reaches the threshold, then any combination of nucleotides for the remaining columns will generate a matching word. If not, it faces two alternative situations. Either the remaining columns may contribute enough to the score to reach the threshold or not. This is checked by summing the maximum score for each of the remaining columns. If this maximum is not sufficient, then the algorithm does not explore this subtree further for any sequence at the leaves have a score that certainly do not reach the threshold. Hence, the algorithm iterates by choosing a new column.

This procedure is iterated until an end has reached when exploring each branch of the search tree. This generation algorithm follows a strategy known as A* strategy.

Once the solution strings have been generated, they are efficiently mapped onto the genome sequence using a adapted genome index [10]. Our strategy could seem costly at first glance, because of the enumeration; however, the results of the comparison with MOODS demonstrate its practical efficiency.

### 2.2 Transforming a PFM into a PWM

To convert PFMs into PWMs, we used the log-odds transformation [13, 2] (Figure 1). Moreover, we used pseudocounts equal to one and background frequencies equal to nucleotidic frequencies in the searched genome.

### 2.3 Relative score threshold

In our work, we decided to use threshold values in the form of relative scores, as it is done in popular cis-regulatory analysis tools such as oPOSSUM [7], since this type of threshold is intuitive to biologists. The user gives the relative score threshold as a percentage p of the PWM matrix score range. Let us denote the score range as $[pwm.min, pwm.max]$. The following formula computes the absolute score threshold:

$$abs_{score} = (pwm.max - pwm.min) * rel_{score} + pwm.min$$

Only the words whose score is above the $abs_{score}$ are searched in the chosen genome.

### 2.4 Data

We used vertebrate and plant matrices from the JASPAR core database [11], version 5.0 (2014). Genome sequences from human16 and maize [12] were downloaded from Ensembl [3]. We used releases GRCh37 (hg19) and AGPv3 (GCA_000005005.5) respectively.

## 3 Results

### 3.1 Comparison with other search tools

Seeking matching binding sites in a target sequence for a given matrix and a given threshold score can be solved exactly, meaning that tools such as MOTIF or MOODS report all occurrences of words matching the matrix with a sufficient score and only those (provided that the sequence lacks undetermined positions - e.g. N). We verified in practice the equality of output positions between MOTIF and MOODS in this general case. The only difference comes from MOTIF ability to prevent reporting matches in regions containing undetermined. Given the correctness of the search, we turn to comparisons in terms of efficacy, and as MOODS has clearly been demonstrated as being much faster than the other used tools for this task, we include solely MOODS in the tests.

We compared the time and memory usages of MOODS and MOTIF for searching complete subsets of JASPAR matrices on the Human and the Maize genomes (Figure 2b, Supplementary Data). To control the number of matching words, the user provides a relative score. The oPOSSUM documentation advised to use values above 80%. For a matrix whose maximum achievable score is smax, a percentage threshold of, say p=85%, means that only words whose score exceeds 85% of smax are accepted as matches. The same threshold was then used for both methods, to ensure that they find the same occurrences, which they do.

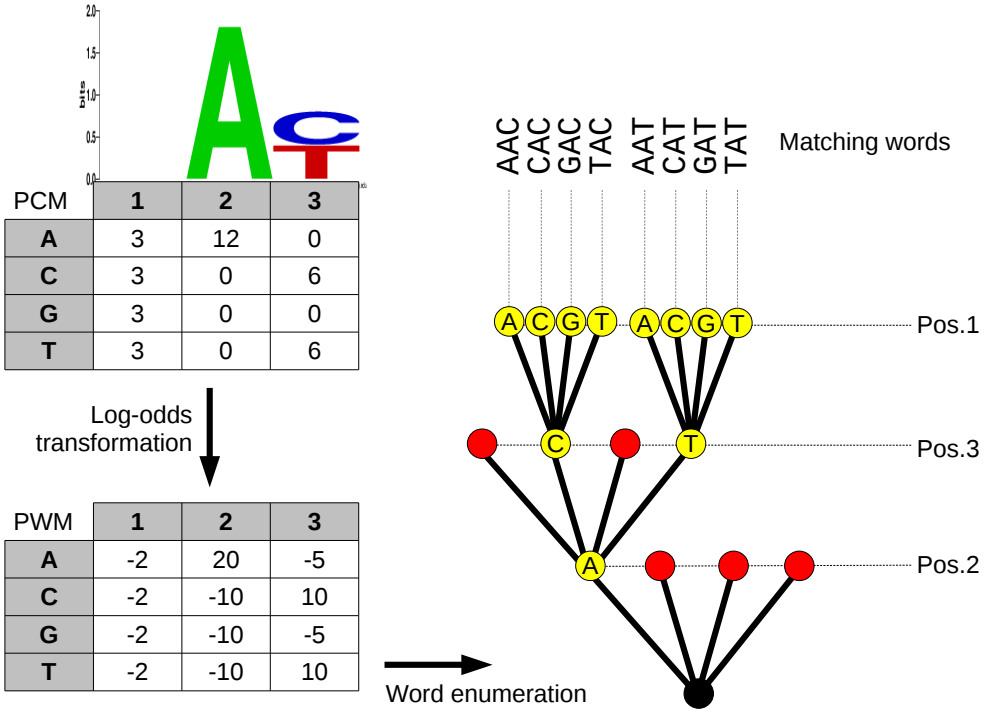



Figure 1: Overview of the word enumeration process in MOTIF. The matrix in the form of a Position Count Matrix (PCM, top left) is first converted into a Position Weight Matrix (PWM, bottom left) using the log-odds transformation. The PCM is also used to derive the information content for each position in the matrix. In this example, consider a threshold value of 25. To generate the strings above this threshold, we build a tree of solutions (right), starting with those positions of higher information content at the lower levels of the tree. This permits to discard the letters at those positions penalizing the score so that it cannot pass the threshold, and subsequently to stop the enumeration for those branches of the tree (red nodes in the tree). This is achieved by constrained programming. In the end of the process, the letters from each string above the threshold are placed back in the correct position to obtain the set of matching words to be mapped onto the genome sequence. For instance, the path reading A at position 2, T at position 3 and A at position 1 yields the matching word AAT.

The run time of MOTIF sums the time of three phases: loading the index, generating possible matching words for a matrix, searching the words in the index and writing the results. MOODS has two phases: first it analyses the matrices to check whether matches are possible according to the threshold, then scans the genome in both orientations and simultaneously writes the results. A notable difference in output: for each occurrence MOODS yields only the genome position and its score, while MOTIF also gives the matching word found. The output volume is thus considerably larger for MOTIF than for MOODS when searching complete genomes.

Clearly, one expects the search time of MOODS to be dominated by the genome scan, while that of MOTIF shall depend on the number of possible matching words and positions. Indeed, the time of MOODS is stable, almost independent of the matrix ( 104 +/- 9 sec for the maize genome with p=85%). MOTIF takes from 0.5 to 53 sec. depending on the matrix, with an average time of 3.49 sec. Its time depends on the matrix, but most of all on the numbers of matches. As illustrated by an example on Human, for two matrices of length 8, with MA0033 MOTIF yields 3 millions matches in 36 sec., while for MA0067 it reports 46000 matches in only 6 sec. Its time depends more on the number of reported positions (and hence on the size of the output), than on the matrix itself.

Despite variable numbers of matching words depending on the matrices, with parameter p=85% on the Human genome, we observed that, in median over all matrices, MOTIF is 106 time faster than MOODS (minimum 2x faster, maximum 180x faster - see Figure 2b). This median speedup increases with p to reach 138x faster when p=90%, and 222x faster when p=95%. The median speed up remains highly favorable to MOTIF even with looser thresholds: for instance 44x faster with p=75% (Supplementary Data) . Logically, MOTIF uses 7 gigabytes of main memory for searching the Human genome whatever the percentage p (70-95%). The memory needed remains in the range of current desktop computers.

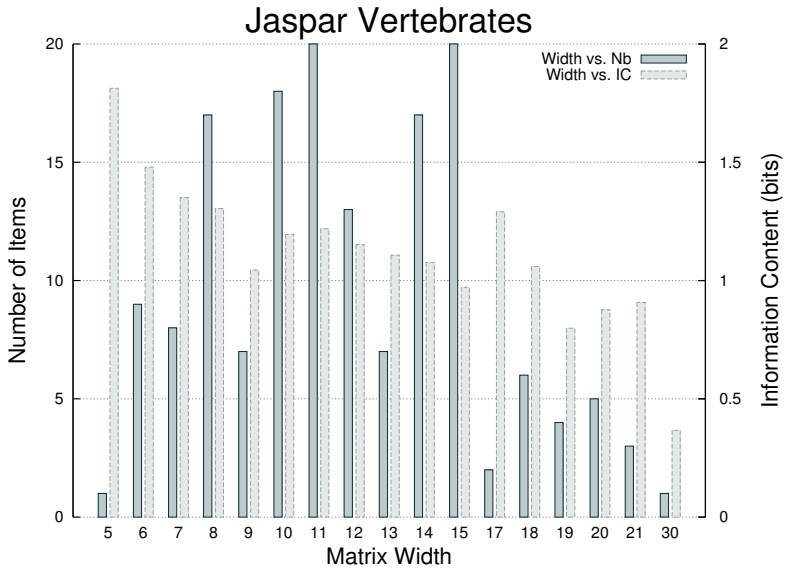
This comparison demonstrates the ability of MOTIF to search for complete sets of PWM motifs on the largest genomes much more efficiently than the fastest available solution and for a large range of parameter values. The important speed up is observed despite the additional output given by MOTIF: indeed, it lists the DNA sequences that match the motif, thereby allowing the user to refine its analysis. Many genome sequences contain numerous undetermined positions (typically coded by an N). In other programs, those are replaced by a predefined or a random nucleotide, thereby generating fake matches. MOTIF handles such regions transparently for the user: in the genome indexes, those are removed and matching regions never overlap them. Those practical features of MOTIF also make the program unique. Altogether, MOTIF shows the power of index-based search algorithms for highly variable nucleotidic motifs.

### 3.2 Web service description

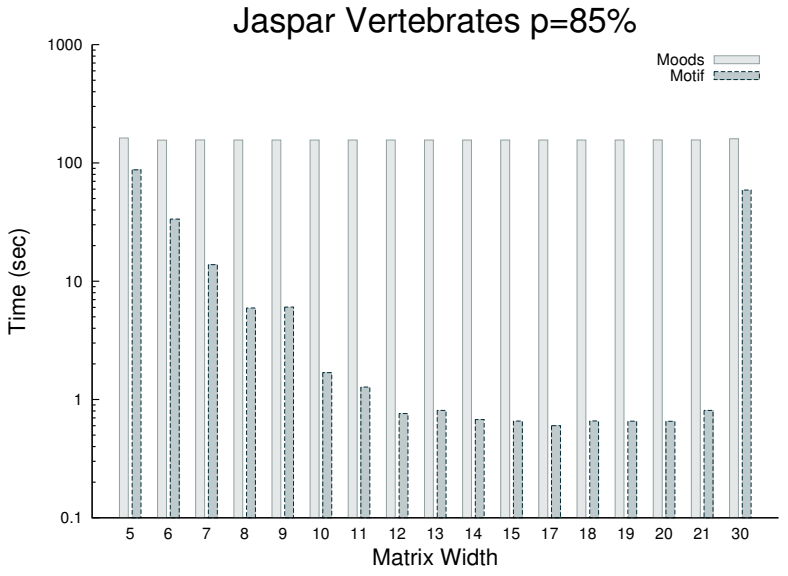
We propose a fast web service to search the JASPAR motifs in large genomes (http://www.atgc-montpellier.fr/motif). The user selects the matrices, the genome, sets the parameter as the percentage of matching words and the strands to search, then gets the results via email. The user may also enter any other matrix interactively. Extensions to the lists of matrices and of searchable sequences are easy to make and could also include a compilation of gene upstream regions for instance, upon user request. To the best of our knowledge, no equivalent services are available for large scale searches of PWM motifs on entire genomes.

## 4 Discussion

In this article we have shown that our new algorithm for predicting TFBS outperforms the fastest existing program known to date. The median speed improvement is 106 fold for a




(a)




(b)

Figure 2: Performance comparison with MOODS. (a) - Distribution of the matrix width and corresponding average information content for all matrices of Transcription Factor Binding Sites from the vertebrate section of JASPAR database. The same figure for JASPAR plant section is in Supplementary Data. (b) - Comparison of median run times between MOODS and MOTIF for searching each vertebrate matrix with a search threshold p=85% in the Human genome. Time scale is logarithmic. MOODS times are almost constant. MOTIF times decrease quickly with the matrix width due to decreasing number of hits found at search stage. MOTIF behaves differently with the unique matrix of width 30 nucleotides due to its very low information content (0.5 bits, see A and Supplementary Data). Similar figures for various thresholds and for plant matrices are provided in Supplementary Data.

threshold value of 85%. We shown that the gain is always favorable to MOTIF when the threshold varies, and the improvement reaches more than 220 fold with stringent threshold values, while it drops to 44 fold when using a loose threshold such as 75%. Only one matrix out of the whole JASPAR collection shows longer running times using MOTIF with lower thresholds (Supplementary Data). In any case, it makes it feasible to scan an entire genome for a whole collection of TF motifs in seconds. Moreover, MOTIF provides a detailed output including the precise location, strand and sequence for each hit, making it more useful for further analysis. We also tackled the problem of matching Ns within the genome sequence, while other programs randomly convert them to any nucleotide. Finally, our service is easy to use, with a user friendly web interface, and the users may decide to use matrix collections provided on our website such as JASPAR, or use their own. While more genomic sequences are being characterized, notably large plant genomes (such as the 22-Gb Loblolly Pine genome [17], our strategy will show stable run times while other algorithms are sequence-length dependent. Moreover, ChIP-Seq has become a routine experimental procedure and may help to refine matrix models in the near future. We could expect those matrices to be more informative (better global IC) and eventually shorter than current ones, which would further enhance the performance of our algorithm.

## Acknowledgements

We thank Vincent Lefort for maintaining the ATGC platform, which hosts the software, and Roderic Guig from the CRG (Barcelona, Spain) for helpful discussions and support.

**Funding** DM was supported by ANR PlasmoExplore, VM by France Gnomique, and ER by the GEM project from Labex NUMev. This work is supported by ANR Institut de Biologie Computationnelle (ANR-11-BINF-0002).

## References

[1] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, June 2006.

[2] G. Z. Hertz and G. D. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15(7):563–577, July 1999.

[3] T. Hubbard, D. Barker, E. Birney, G. Cameron, Y. Chen, L. Clark, T. Cox, J. Cuff, V. Curwen, T. Down, R. Durbin, E. Eyras, J. Gilbert, M. Hammond, L. Huminiecki, A. Kasprzyk, H. Lehvaslaiho, P. Lijnzaad, C. Melsopp, E. Mongin, R. Pettett, M. Pocock, S. Potter, A. Rust, E. Schmidt, S. Searle, G. Slater, J. Smith, W. Spooner, A. Stabenau, J. Stalker, E. Stupka, A. Ureta-Vidal, I. Vastrik, and M. Clamp. The Ensembl genome database project. *Nucleic Acids Research*, 30(1):38–41, January 2002.

[4] Arttu Jolma, Jian Yan, Thomas Whitington, Jarkko Toivonen, KazuhiroR. Nitta, Pasi Rastas, Ekaterina Morgunova, Martin Enge, Mikko Taipale, Gonghong Wei, Kimmo Palin, JuanM. Vaquerizas, Renaud Vincentelli, NicholasM. Luscombe, TimothyR. Hughes, Patrick Lemaire, Esko Ukkonen, Teemu Kivioja, and Jussi Taipale. DNA-Binding Specificities of Human Transcription Factors. *Cell*, 152(12):327–339, January 2013.

[5] Janne Korhonen, Petri Martinmki, Cinzia Pizzi, Pasi Rastas, and Esko Ukkonen. MOODS: fast search for position weight matrix matches in DNA sequences. *Bioinformatics*, 25(23):3181–3182, January 2009.

[6] Ivan V. Kulakovskiy, Yulia A. Medvedeva, Ulf Schaefer, Artem S. Kasianov, Ilya E. Vorontsov, Vladimir B. Bajic, and Vsevolod J. Makeev. HOCOMOCO: a comprehensive collection of human transcription factor binding sites models. *Nucleic Acids Research*, 41(D1):D195–D202, January 2013.

[7] Andrew T. Kwon, David J. Arenillas, Rebecca Worsley Hunt, and Wyeth W. Wasserman. oPOSSUM-3: Advanced Analysis of Regulatory Motif Over-Representation Across Genes or ChIP-Seq Datasets. *G3: Genes|Genomes|Genetics*, 2(9):987–1002, September 2012.

[8] P. J. Mitchell and R. Tjian. Transcriptional regulation in mammalian cells by sequence-specific DNA binding proteins. *Science*, 245(4916):371–378, July 1989.

[9] Carl O. Pabo and Robert T Sauer. Transcription Factors: Structural Families and Principles of DNA Recognition. *Annual Review of Biochemistry*, 61(1):1053–1095, 1992.

[10] Nicolas Philippe, Mikael Salson, Thérèse Commes, and Eric Rivals. CRAC: an integrated approach to the analysis of RNA-seq reads. *Genome Biology*, 14(3):R30, 2013.

[11] Elodie Portales-Casamar, Supat Thongjuea, Andrew T. Kwon, David Arenillas, Xiaobei Zhao, Eivind Valen, Dimas Yusuf, Boris Lenhard, Wyeth W. Wasserman, and Albin Sandelin. JASPAR 2010: the greatly expanded open-access database of transcription factor binding profiles. *Nucleic Acids Research*, 38(Database issue):D105–D110, January 2010.

[12] Patrick S. Schnable, Doreen Ware, Robert S. Fulton, Joshua C. Stein, Fusheng Wei, Shiran Pasternak, Chengzhi Liang, Jianwei Zhang, Lucinda Fulton, Tina A. Graves, Patrick Minx, Amy Denise Reily, Laura Courtney, Scott S. Kruchowski, Chad Tomlinson, Cindy Strong, Kim Delehaunty, Catrina Fronick, Bill Courtney, Susan M. Rock, Eddie Belter, Feiyu Du, Kyung Kim, Rachel M. Abbott, Marc Cotton, Andy Levy, Pamela Marchetto, Kerri Ochoa, Stephanie M. Jackson, Barbara Gillam, Weizu Chen, Le Yan, Jamey Higginbotham, Marco Cardenas, Jason Waligorski, Elizabeth Applebaum, Lindsey Phelps, Jason Falcone, Krishna Kanchi, Thynn Thane, Adam Scimone, Nay Thane, Jessica Henke, Tom Wang, Jessica Ruppert, Neha Shah, Kelsi Rotter, Jennifer Hodges, Elizabeth Ingenthron, Matt Cordes, Sara Kohlberg, Jennifer Sgro, Brandon Delgado, Kelly Mead, Asif Chinwalla, Shawn Leonard, Kevin Crouse, Kristi Collura, Dave Kudrna, Jennifer Currie, Ruifeng He, Angelina Angelova, Shanmugam Rajasekar, Teri Mueller, Rene Lomeli, Gabriel Scara, Ara Ko, Krista Delaney, Marina Wissotski, Georgina Lopez, David Campos, Michele Braidotti, Elizabeth Ashley, Wolfgang Golser, HyeRan Kim, Seunghee Lee, Jinke Lin, Zeljko Dujmic, Woojin Kim, Jayson Talag, Andrea Zuccolo, Chuanzhu Fan, Aswathy Sebastian, Melissa Kramer, Lori Spiegel, Lidia Nascimento, Theresa Zutavern, Beth Miller, Claude Ambroise, Stephanie Muller, Will Spooner, Apurva Narechania, Liya Ren, Sharon Wei, Sunita Kumari, Ben Faga, Michael J. Levy, Linda McMahan, Peter Van Buren, Matthew W. Vaughn, Kai Ying, Cheng-Ting Yeh, Scott J. Emrich, Yi Jia, Ananth Kalyanaraman, An-Ping Hsia, W. Brad Barbazuk, Regina S. Baucom, Thomas P. Brutnell, Nicholas C. Carpita, Cristian Chaparro, Jer-Ming Chia, Jean-Marc Deragon, James C. Estill, Yan Fu, Jeffrey A. Jeddeloh, Yujun Han, Hyeran Lee, Pinghua Li, Damon R. Lisch, Sanzhen Liu, Zhijie Liu, Dawn Holligan Nagel, Maureen C. McCann, Phillip SanMiguel, Alan M. Myers, Dan Nettleton, John Nguyen, Bryan W. Penning, Lalit Ponnala, Kevin L. Schneider, David C. Schwartz, Anupma Sharma, Carol Soderlund, Nathan M. Springer, Qi Sun, Hao Wang, Michael Waterman, Richard Westerman, Thomas K. Wolfgruber, Lixing Yang, Yeisoo Yu, Lifang Zhang, Shiguo Zhou, Qihui Zhu, Jeffrey L. Bennetzen, R. Kelly Dawe, Jiming Jiang, Ning Jiang, Gernot G. Presting, Susan R. Wessler, Srinivas Aluru, Robert A. Martienssen,

Sandra W. Clifton, W. Richard McCombie, Rod A. Wing, and Richard K. Wilson. The B73 Maize Genome: Complexity, Diversity, and Dynamics. *Science*, 326(5956):1112–1115, November 2009.

[13] Rodger Staden. Methods for calculating the probabilities of finding patterns in sequences. *Bioinformatics*, 5(2):89–96, April 1989.

[14] Anne-Laure Todeschini, Adrien Georges, and Reiner A. Veitia. Transcription factors: specific DNA binding and specific gene regulation. *Trends in Genetics*, 30(6):211–219, June 2014.

[15] Wyeth W. Wasserman and Albin Sandelin. Applied bioinformatics for the identification of regulatory elements. *Nature Reviews Genetics*, 5(4):276–287, April 2004.

[16] E. Wingender, P. Dietze, H. Karas, and R. Knppel. TRANSFAC: A Database on Transcription Factors and Their DNA Binding Sites. *Nucleic Acids Research*, 24(1):238–241, January 1996.

[17] Aleksey Zimin, Kristian A. Stevens, Marc W. Crepeau, Ann Holtz-Morris, Maxim Koriabine, Guillaume Marais, Daniela Puiu, Michael Roberts, Jill L. Wegrzyn, Pieter J. de Jong, David B. Neale, Steven L. Salzberg, James A. Yorke, and Charles H. Langley. Sequencing and Assembly of the 22-Gb Loblolly Pine Genome. *Genetics*, 196(3):875–890, March 2014.

# Supplemental Data

## 4.1 A vertebrate matrix with very weak information content

We observe that one vertebrate matrix induces much longer search times than all other matrices, because it generates huge numbers of potential matching words especially with low thresholds (see result graph with $p = 80\%$). This matrix (JASPAR ID MA0068) has a width of 30. Usually the search time of our tool decreases with the width. Investigating the reason of this unusual behavior, we looked at the information content of that matrix: the matrix MA0068 is the only matrix of that width, and the only matrix in vertebrate and even in plant sections with an information content way below 0.5.

The logo view of matrix MA0068 is displayed below.

The evolution of Motif search time for MA0068 in the Human genome when $p$ varies from 70 to 95% suggests that a threshold value of at least 90% is reasonable for this weak matrix.

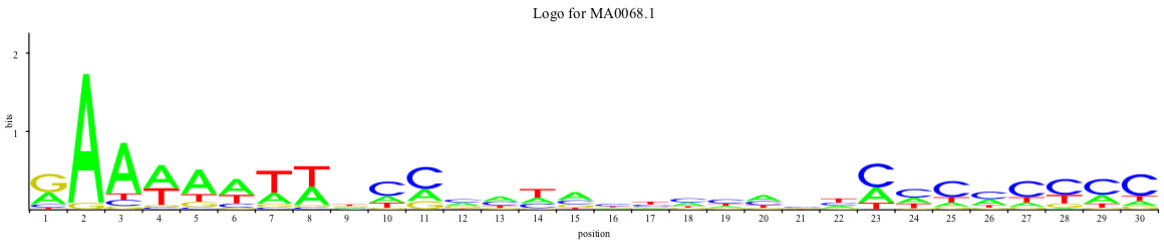



Figure 3: LOGO visualisation of JASPAR matrix MA0068. The height of the stack of letter in each column gives the information content in bits of that column. Only one out of thirty columns reaches a value of 1.

### 4.2 Search time comparison

We compared the running time of Moods [5] and Motif for searching each matrix on a large genome and let the main parameter $p$ vary with values in $[70, 95]\%$ with a step of 5. The default value of $p$ is 85%, which is the value advised by JASPAR [11].

As in the article, to illustrate the difference in running times between both tools, we plot for each value of $p$ a histogram of average search times for both tools with respect to matrix width.

Figure 5 displays the histograms for searching the reference *Zea mays* genome using all JASPAR plant matrices for $p = 70$ and 75, $p = 80$ and 85, and $p = 90$ and 95, respectively. Figure 6 plots the pendant histograms for searching the Human genome with all JASPAR vertebrate matrices (the figure for $p = 85\%$ is also shown in Figure 2 the manuscript).

### 4.3 With JASPAR plant matrices

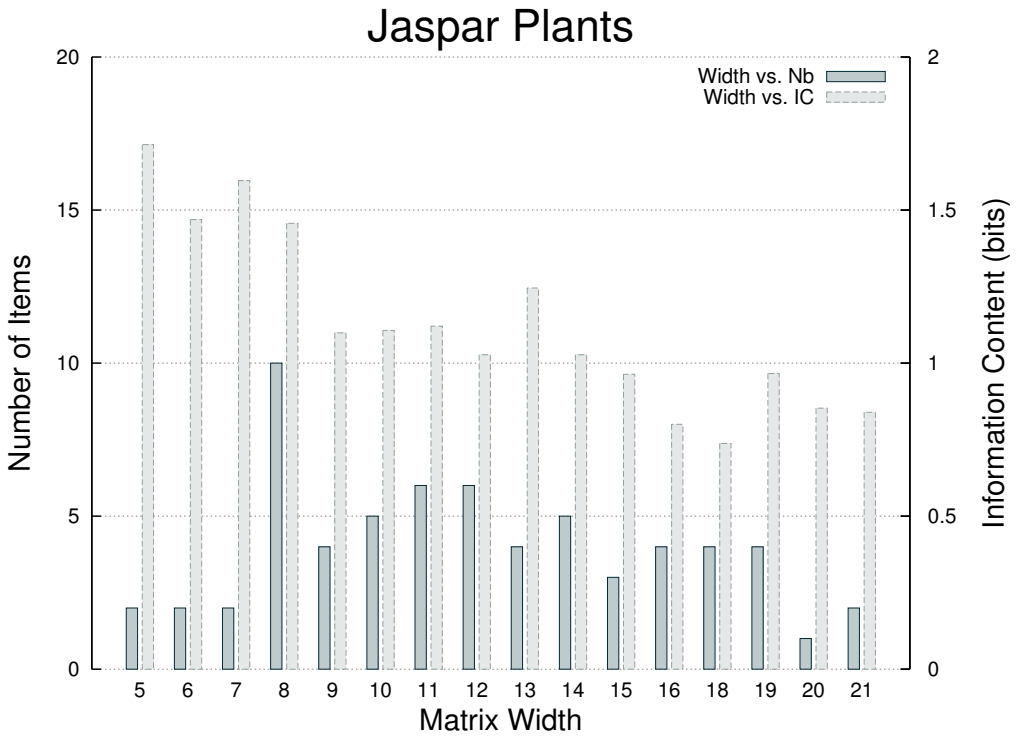
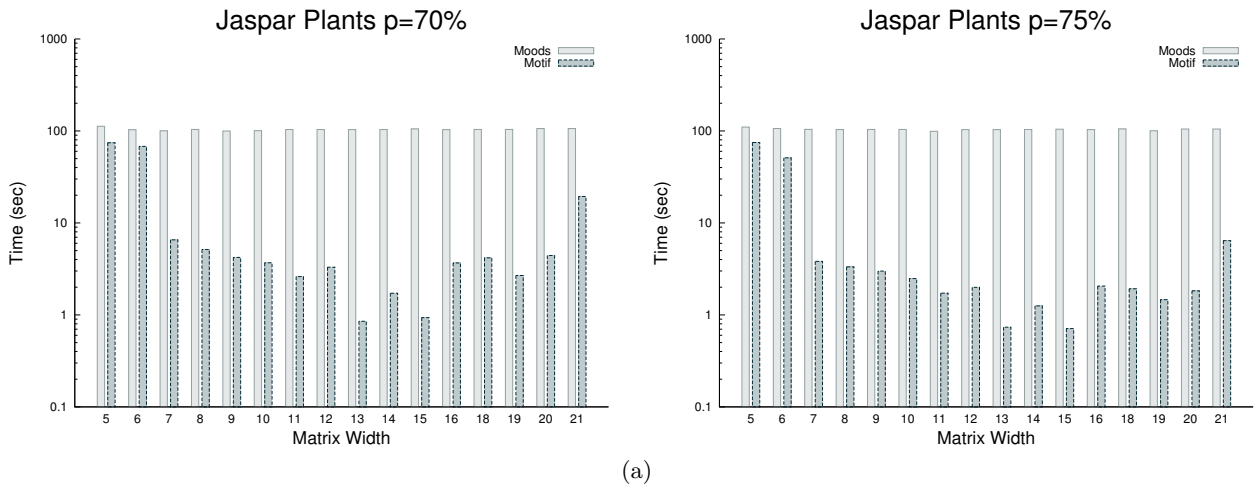
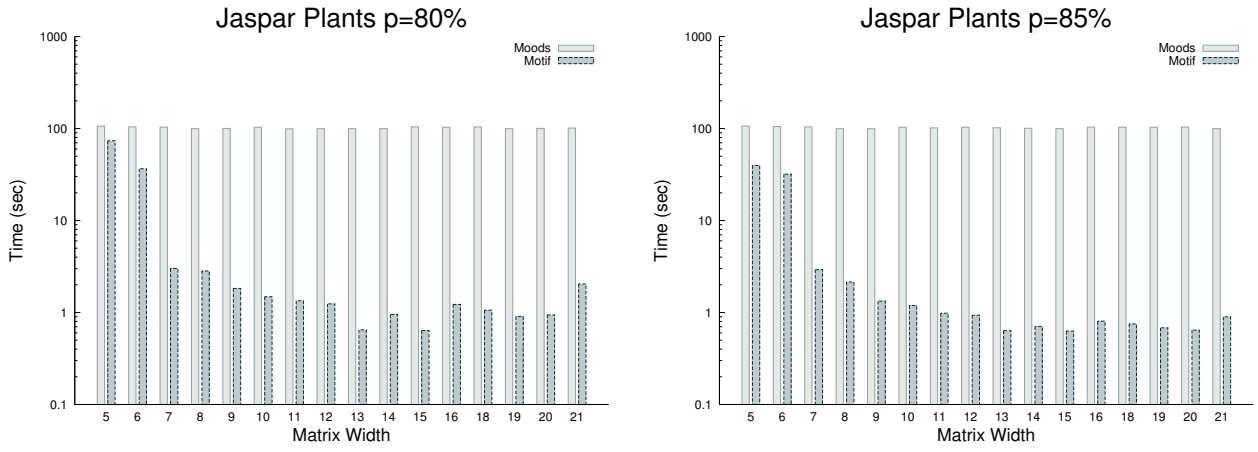


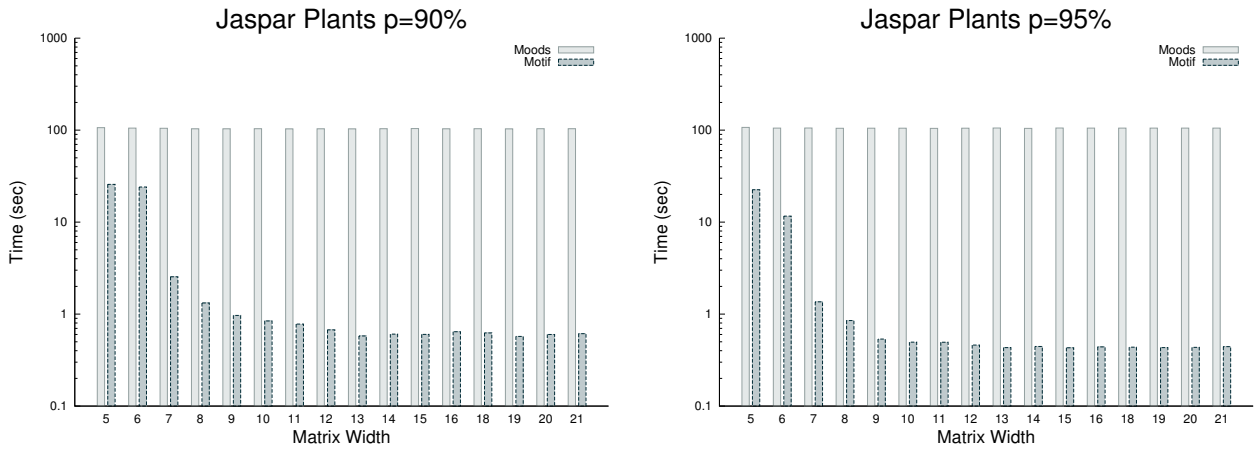

Figure 4: Distribution of the matrix width and corresponding average information content for all matrices of Transcription Factor Binding Sites from the plant section of JASPAR database [11].




(a)



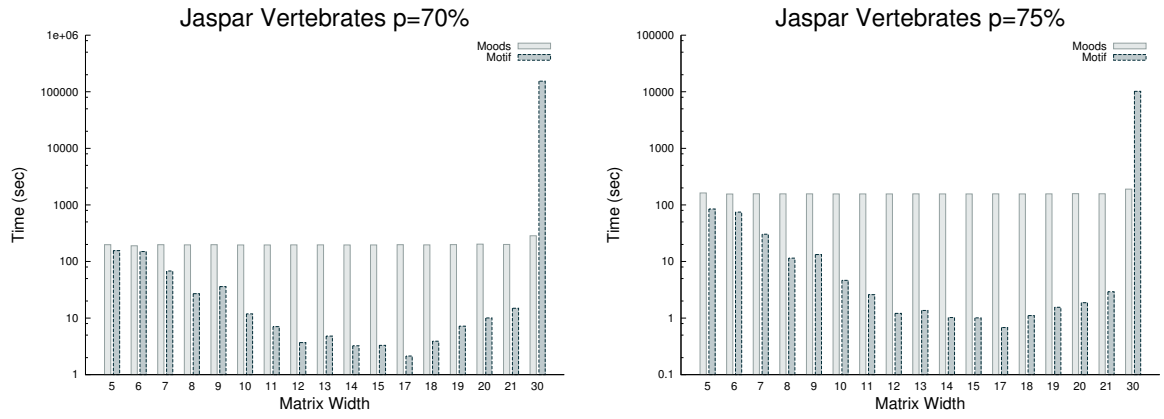

(b)



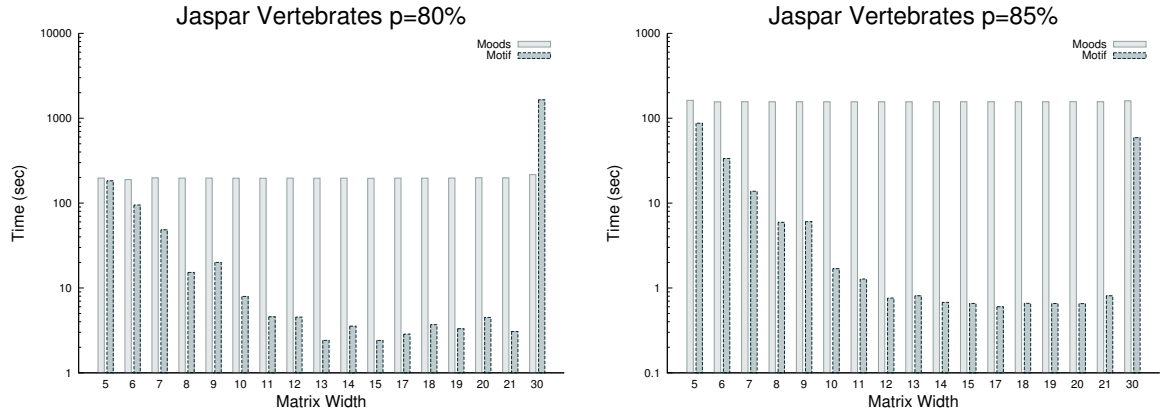

(c)

Figure 5: Median running times of Moods and Motif for searching JASPAR plant matrices in the *Zea mays* genome with parameter (a) $p = 70$ or 75%, (b) $p = 80$ or 85%, (c) $p = 90$ or 95%.

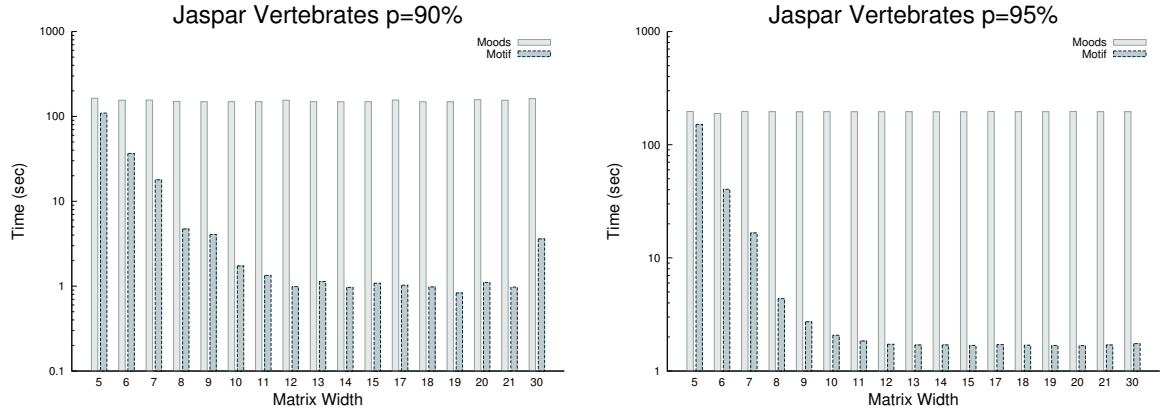
## 4.4 With JASPAR vertebrate matrices




(a)




(b)




(c)

Figure 6: Median running times of Moods and Motif for searching JASPAR vertebrate matrices in the Human genome with parameter (a) $p = 70$ or 75%, (b) $p = 80$ or 85%, (c) $p = 90$ or 95%.