OPEN ACCESS Freely available online

PLOS ONE

CisMiner: Genome-Wide *In-Silico* Cis-Regulatory Module Prediction by Fuzzy Itemset Mining

Carmen Navarro[1]*, Francisco J. Lopez[2], Carlos Cano[1], Fernando Garcia-Alcalde[3], Armando Blanco[1]

1 Department of Computer Science and AI, University of Granada, Granada, Spain, **2** Andalusian Human Genome Sequencing Centre (CASEGH), Medical Genome Project (MGP), Sevilla, Spain, **3** Max Planck Institute for Infection Biology, Berlin, Germany

Abstract

Eukaryotic gene control regions are known to be spread throughout non-coding DNA sequences which may appear distant from the gene promoter. Transcription factors are proteins that coordinately bind to these regions at transcription factor binding sites to regulate gene expression. Several tools allow to detect significant co-occurrences of closely located binding sites (cis-regulatory modules, CRMs). However, these tools present at least one of the following limitations: 1) scope limited to promoter or conserved regions of the genome; 2) do not allow to identify combinations involving more than two motifs; 3) require prior information about target motifs. In this work we present CisMiner, a novel methodology to detect putative CRMs by means of a fuzzy itemset mining approach able to operate at genome-wide scale. CisMiner allows to perform a blind search of CRMs without any prior information about target CRMs nor limitation in the number of motifs. CisMiner tackles the combinatorial complexity of genome-wide cis-regulatory module extraction using a natural representation of motif combinations as itemsets and applying the Top-Down Fuzzy Frequent- Pattern Tree algorithm to identify significant itemsets. Fuzzy technology allows CisMiner to better handle the imprecision and noise inherent to regulatory processes. Results obtained for a set of well-known binding sites in the *S. cerevisiae* genome show that our method yields highly reliable predictions. Furthermore, CisMiner was also applied to putative in-silico predicted transcription factor binding sites to identify significant combinations in *S. cerevisiae* and *D. melanogaster*, proving that our approach can be further applied genome-wide to more complex genomes. CisMiner is freely accesible at: http://genome2.ugr.es/cisminer. CisMiner can be queried for the results presented in this work and can also perform a customized cis-regulatory module prediction on a query set of transcription factor binding sites provided by the user.

Citation: Navarro C, Lopez FJ, Cano C, Garcia-Alcalde F, Blanco A (2014) CisMiner: Genome-Wide *In-Silico* Cis-Regulatory Module Prediction by Fuzzy Itemset Mining. PLoS ONE 9(9): e108065. doi:10.1371/journal.pone.0108065

Editor: Leonardo Mariño-Ramírez, National Institutes of Health, United States of America

Received July 3, 2014; **Accepted** August 25, 2014; **Published** September 30, 2014

Data Availability: The authors confirm that all data underlying the findings are fully available without restriction. All relevant data are within the paper and its Supporting Information files.

Funding: This work has been carried out as part of projects PI-0710-2013 of J. A., Sevilla and TIN2013-41990-R of DGICT, Madrid. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* Email: cnluzon@decsai.ugr.es

Introduction

An organism's DNA encodes the information required for each cell to function. However, a complete description of the DNA sequence of an organism is not enough to reconstruct it. Not only genes (i.e. coding DNA) hold relevant information, but also the rest of non-coding DNA, which orchestrates how each element is related to the rest: under which conditions each gene product is made, and which role it plays in the complex machinery of the cell. There are many steps in the pathway leading from DNA to protein. The initiation of RNA transcription is a very important step in such pathway [1].

Eukaryotic gene control regions consist of a promoter region plus a set of regulatory DNA sequences. Transcription factors (TFs) are regulatory proteins that bind at these regions to specific sequences called transcription factor binding sites (TFBSs) forming complexes that are essential to the initiation of gene transcription. In addition, this TF-DNA interaction is usually coordinated forming *cis*-regulatory modules (CRMs) [2].

Many different approaches address *in silico* CRM detection. However, CRM detection has strong performance limitations due to its combinatorial complexity [3]. Overall, three conceptually different classes of methods can be roughly identified according to an increasing genomic scope [4]: 1) CRM scanners, 2) CRM builders and 3) CRM genome screeners.

CRM scanners search for sequences that satisfy a strictly defined CRM model. Their goal is usually to further study well-characterized problems, and the user is required to provide a detailed specification of the studied CRM. Therefore, their application is limited to well-known problems and many parameters are usually required, such as the expected distance between TFBSs, a background model for the sequences, number of target TFBSs, a reduced set of target TFBSs or a window size. Cister [5], Cluster-Buster [6] or Stubb [7] belong to this category. Due to their specificity, their execution times tend to be shorter than other methods with a broader scope.

CRM builders extend the scope of the CRM prediction by reducing the number of constraints for the target CRM model. They try to assemble CRMs looking for similar features in a reduced set of related sequences. These methods reduce the sequence search space by two main approaches: 1) limiting the study to gene promoter regions of co-expressed or related genes of

interest [8, 9]; or 2) focusing on evolutionary conserved regions [10–12]. The first type of approaches could miss regulatory elements in less obvious locations, such as those located in introns and far upstream or downstream of genes. Indeed, genome-wide chromatin immunoprecipitation experiments [13] have reported that a significant proportion of TFBSs do not lie in regions immediately upstream of known protein-coding genes [14–17]. In addition, comparative genomics revealed that many regulatory elements involved in early vertebrate development lie far from the gene they are thought to regulate [10, 18–20]. The second type of approaches are based on the evolutionary conservation of regulatory sequences. However, the regulatory sequences have to be similar enough to be aligned, which is often not the case when compared sequences are not closely related. Moreover, experiments on mammalian and *Drosophila* species have shown that between one and two-thirds of identified regulatory sequences are not conserved even between closely related species [10, 21–23]. Approaches like INSECT [24] and CORECLUST [25] optionally combine these two techniques to reduce the search space.

According to the description in the literature [4], **CRM genome screeners** search through complete genomes for CRMs. In contrast to the previous approaches, genome screeners do not make any assumptions regarding the target CRMs. Therefore, they present a broader applicability but tackle a more complex problem. However, although some methods are considered CRM genome screeners in [4], a genome-wide screening tool that encompasses the whole non-coding DNA of an organism without any restriction has not yet been made available. Some approaches, such as D-light [26], do not restrict the number of genes but limit the screening to the promoter regions of these genes, not considering the rest of the non-coding genome. Approaches like COPS [27] search for co-ocurring TFs in sequences known to be bound *in vivo*, therefore requiring previous knowledge on target TFs and experimental evidence of their binding sites. Other approaches like TraFaC [28], PreMod [11] or EEL [12], also reduce the search space to a set of orthologous sequences or co-regulated genes, and therefore could be considered CRM builders. CisMiner, the methodology we present in this work, can be framed into the CRM genome screeners category.

Independently of their scope, all the aforementioned approaches suffer the computational complexity of CRM detection, which is not only increased by the total length of the sequences to screen, but also by the number of putative TFBSs detected and the size of the target modules. In this sense, some approaches reduce the search space by limiting the number of cooperating transcription factors, usually looking for pairs of co-ocurring transcription factors [29, 30]. Although these are easier to identify and have been shown to have biological significance, CRMs can encompass larger sets of transcription factors which relate in a more sophisticated manner [2]. These complex CRMs are overlooked with such approaches.

Perhaps because of its combinatorial complexity, specially when applied genome-wide, the vast majority of available approaches for computational CRM discovery restrict the search space (to a set of co-regulated genes and/or orthologous sequences) or the dimensionality of the problem (number or positioning of TFBSs), overlooking larger CRMs or those located in other non-coding regions of the genome. With the currently available approaches, any researcher aiming to obtain a set of putative cis-regulatory modules in a query organism with no prior information about which transcription factors are involved in the process will need a specific set of motifs as an input.

This might not be likely if the researcher does not have any prior knowledge. Thus, the probability of selecting the adequate subset of TFBSs would be equivalent to the probability of picking a random subset of the given set of TFBSs in an organism, which is very small. The same applies for approaches that use orthologous sequences to a certain gene or a set of co-regulated genes. These approaches are useful for the study of already known processes. However, it is very unlikely that a researcher trying to explore a certain organism without prior knowledge will be able to provide a set of specific motifs and sequences useful for extracting knowledge in a given organism.

On the other hand, CRM prediction tools are usually coupled with TFBS detection tools. Therefore, CRM prediction usually requires a set of position weight matrices (PWMs) as input to first apply a TFBS detection tool to screen the query sequences for a set of putative TFBSs. In this case, the overall performance of CRM genome screeners depends heavily on the quality of the predictions made by the TFBSs detection tool. CisMiner, the tool we propose, is able to perform CRM prediction by either using a computational TFBS prediction tool for whole genome sreening, or a set of already predicted or known TFBSs provided by the user.

Furthermore, information about the exact location and length of regulatory regions is imprecise and uncertain. It is known that CRMs can span hundreds of base pairs, although their exact length has not been assessed. Not only the location and length of CRMs, but also the information about the positions where transcription factors bind to the genome, are still vague and inaccurate. *In-silico* tools for TFBS prediction sometimes surpass the reasonable amount of false positives due to the probabilistic and biological complexity of this problem. The sought sequences are indeed very short ($10-30bp$) compared to the size of the scanned genomes, and the appearance of sequences which are truly similar to those of real TFBSs but do not represent a regulatory function must be expected as well. This lack of transcription factor sequence specificity may suggest that more complex rules and mechanisms govern the regulation process influenced by transcription factor binding activity [2].

Any computational CRM prediction approach must therefore take these problems into consideration and handle the imprecision and uncertainty unavoidably present in the data. However, although fuzzy techniques are known to outperform classical crisp techniques when dealing with imprecise and noisy data, these approaches are barely used in this field.

In this work we present CisMiner, a fuzzy-based genome-wide CRM screener which overcomes some of the mentioned limitations by allowing to scan whole non-coding genomes for significant combinations of any number of TFBSs. Given a set of TFBSs, CisMiner implements a fuzzy clustering of closely-located TFBSs and analyze these fuzzy sets to obtain combinations of TFBSs which co-occur significantly using the Top-Down Fuzzy Frequent-Pattern Tree algorithm.

To the extent of our knowledge, there are not any available approaches that encompass the whole non-coding genome of an organism and allow a search for highly dimensional CRMs. Therefore, the area of application of CisMiner differs greatly to those of the rest of mentioned approaches. CisMiner is based on performing a fuzzy frequent itemset mining on a large set of fuzzy clusters in order to find significant patterns affecting an organism genome-wide. The tailored sensitivity of CRM predictors based on related sequences may be useful for studying the specifics of previously established mechanisms, although the application of such tools genome-wide may not be suitable due to performance restrictions and the lack of enough prior knowledge. CisMiner is, in this sense, a unique approach for obtaining reliable putative

CRMs for whole-genome studies with no prior assumptions about genes or transcription factors involved.

This work is organized as follows. First, the proposed methodology is described in detail. Second, obtained results for *Saccharomyces cerevisiae* and *Drosophila melanogaster* are presented and discussed. We show that many of the obtained relations between TFs are supported by previous scientific evidence. In addition, confident new putative cis-regulatory modules have been obtained, contributing to the discovery of new regulatory relations. Finally, conclusions and future work are discussed. CisMiner is freely accessible at http://genome2.ugr.es/cisminer.

Methods

Methodology overview

CisMiner implements a data analysis pipeline that takes a set of transcription factor binding sites (TFBSs) as input and provides a set of significant co-ocurrences of any number of TFBSs as output. To this end, the following steps are performed. Given a set of TFBSs, fuzzy clusters of closely-located TFBSs are detected genome-wide. These fuzzy sets are included as itemsets in a fuzzy transactional database, which is in turn mined to obtain combinations of TFBSs which co-occur significantly. The Fuzzy Frequent-Pattern Tree (Fuzzy FP-Tree) algorithm, a fuzzy frequent itemset mining algorithm developed by the authors, is applied to this end, since it has previously shown a good performance for very large datasets [31]. The set of TFBSs used as input can either be predicted in-silico or in-vivo, allowing the user to couple this methodology to any in-silico TFBSs prediction tool to perform a prior genome-wide search for a set of putative TFBSs, which can then be given to CisMiner as an input to discover CRMs. An outline of the procedure is shown in Figure 1.

Clustering of TFBSs and fuzzy transactional database construction

In order to be able to extract significant groups of TFs that form putative cis-regulatory modules (CRMs), the first step taken by CisMiner is to perform a fuzzy clustering of closely located TFBSs and model this set of clusters as a fuzzy transactional database.

Fuzzy set theory was proposed by Zadeh in 1965 to mathematically model the imprecision inherent to some concepts [32]. Briefly, fuzzy set theory allows an object to partially belong to a set with a membership degree between 0 and 1.

Frequent Itemset Mining was proposed by Agrawal in 1993, as an algorithm for extracting frequent itemsets from large databases [33]. Since then, a large number of algorithms have been proposed for frequent-itemset mining [34]. Given a transactional database where each transaction is a set of *items*, the aim of these techniques is to find a set of expressions of the form $\{x_1,x_2,x_3,...,x_i\}$, where each x_i represents an item. This expression is called *itemset*. The probability that a given itemset occurs in the data base is called the *support* of the itemset. If the support of an itemset is greater than a user-specified threshold, then the itemset is said to be *frequent*. Thus, Frequent Itemset Mining algorithms aim to extract itemsets from a database with support greater than some user-specified threshold.

Frequent Itemset Mining algorithms have featured many applications that enable researchers to unveil hidden patterns in large amounts of data [35]. However, biological data are usually uncertain and imprecise. In order to be able to reflect this uncertainty, fuzzy technology has been incorporated to the

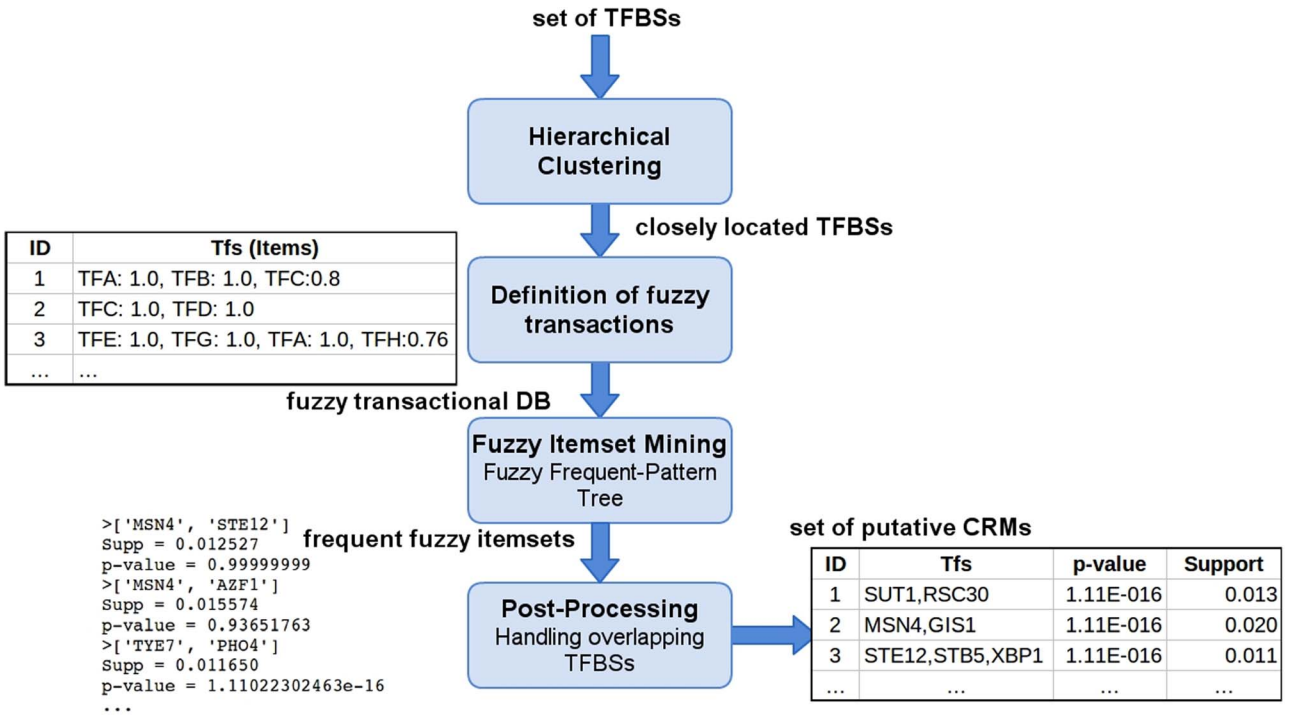


Figure 1. Outline of the CisMiner procedure. Diagram of the main steps of the CisMiner procedure. Given a set of TFBSs, the process starts by performing a fuzzy hierarchical clustering to obtain a set of closely located TFBSs. The result of this step is a fuzzy transactional database, which will then be mined by a Fuzzy Frequent Itemset Mining algorithm (Fuzzy Frequent-Pattern Tree) to obtain a set of frequent fuzzy itemsets. Finally, a postprocessing takes place in order to handle overlapping TFBSs that appear in each frequent itemset. As a result, a set of putative CRMs, along with their estimated p-value and their fuzzy support, is given.
doi:10.1371/journal.pone.0108065.g001

Frequent Itemset Mining philosophy in the Fuzzy FP-Tree algorithm applied in this work.

Connecting fuzzy technology with frequent itemset mining allows us to incorporate uncertainty and imprecision to our knowledge model. In this sense, fuzzy itemsets are also expressions of the form $\{x_1,x_2,...,x_i\}$, but in this case, each x_i is accompanied by a value in [0,1] which defines its membership degree to the itemset. Fuzzy support measures the frequency of the itemset.

CisMiner combines fuzzy theory with frequent itemsets in order to increase their capacity of modelling the uncertainty present in biological data, and, in particular, TFBS binding location data. The fuzzy clusters of closely located TFBSs are modelled as itemsets in a fuzzy transactional database. In order to build such database, a group-average hierarchical clustering (the implementation of the hierarchical clustering method was obtained from the python-hcluster package release 0.2.0-1).algorithm is run over the set of TFBS locations.

An upper-threshold of $300bp$ was set for stopping the cluster aggregation. That is, we seek groups of TFBSs which span around $\sim 300bp$ in the genome, assuming that regulatory modules are generally a few hundred base pairs in length [36].

Once the list of clusters was obtained, a *fuzzy* transaction was defined for each cluster. The membership degree function for each TFBS in each cluster was defined as a trapezoidal function, as it is shown in figure 2, with the following parameters: a centroid C was obtained for each cluster as the median value of the position of the included TFBSs. Then, the constant region of the trapezoidal function was set from $C-150$ to $C+150$. A linear increasing function in $[C-250,C-150]$ and a decreasing function in $[C+250,C+150]$ were defined to set membership degrees at the fuzzy borders of each cluster.

Once the membership degree functions were defined, a set of fuzzy transactions were built. Figure 2 shows an outline of this procedure.

Mining the frequent fuzzy itemsets

After the fuzzy transactional database has been built, CisMiner proceeds to obtain significant sets of co-occurring TFs (i.e. putative CRMs). The fuzzy frequent itemset mining procedure enables CisMiner to remove non-significant TF clusters at a genome-wide scale. We implemented a fuzzy frequent itemset mining algorithm called the Top-Down Fuzzy Frequent Pattern-Growth algorithm, which has been developed by the authors in a previous work [31].

Briefly, this procedure works as follows (for a more exhaustive description of the Fuzzy FP-Tree data structure generation and traversing, see [31]). Initially, the algorithm scans the transaction database in order to get a sorted list of all the frequent items, i.e. items with support greater than a specified threshold. An item x_i represents one TFBS and belongs to a transaction of the form $\{x_1,x_2,x_3,...,x_n\}$ with a certain membership degree. The aim of the frequent itemset mining procedure is to extract significant itemsets, i.e. collections of items that appear frequently among the transactions in the database. After the items with higher support than the specified threshold have been selected, the items in this list along with their membership degrees for each transaction are introduced into the Fuzzy FP-Tree data structure. The efficiency of this procedure relies on the use of this tree, since it compiles all the information the algorithm needs from the transactions.

Items in each transaction that are present in the frequent items list are inserted as nodes into the Fuzzy FP-Tree according to their position in the frequent item list. If two transactions share their first frequent items they will share the same upper path to the root node.

For each item I, all its nodes are linked by a $side-links$ list. A vector associated to each node stores the membership degree of the transactions that belong to the corresponding item. In addition, a header table H is built so each row stores the information associated to an item I: (Item, membership degrees, $side-links$). This table helps locating the nodes that correspond to each item in the Fuzzy FP-tree and to compute the fuzzy support of each itemset.

Once the Fuzzy FP-Tree and the header table H are generated, the tree is traversed in a top-down manner in order to obtain the set of frequent itemsets. Entries in H are considered one by one. For each item I in the H table the tree is traversed in a down-top order, starting at the nodes labeled with I. These nodes can be reached following the $side-links$ list. Each node needs a membership degree vector that keeps the minimum membership degree between the starting and the current node. This is crucial because it ensures that modifications of these vectors at upper levels do not affect the processing of lower level nodes.

Fuzzy support for each itemset was calculated as described in [37]. In addition, a p-value was calculated in order to complement the frequency value provided by the support measure. The procedure reported in [38] for the p-value computation was adapted for the fuzzy case. The null model for the calculation of this p-value represents the uninteresting situation in which no item associations are present, i.e. in which items occur independently from each other in transactions. Thus, the p-value represents the probability of the itemset to be surprising under the null-model.

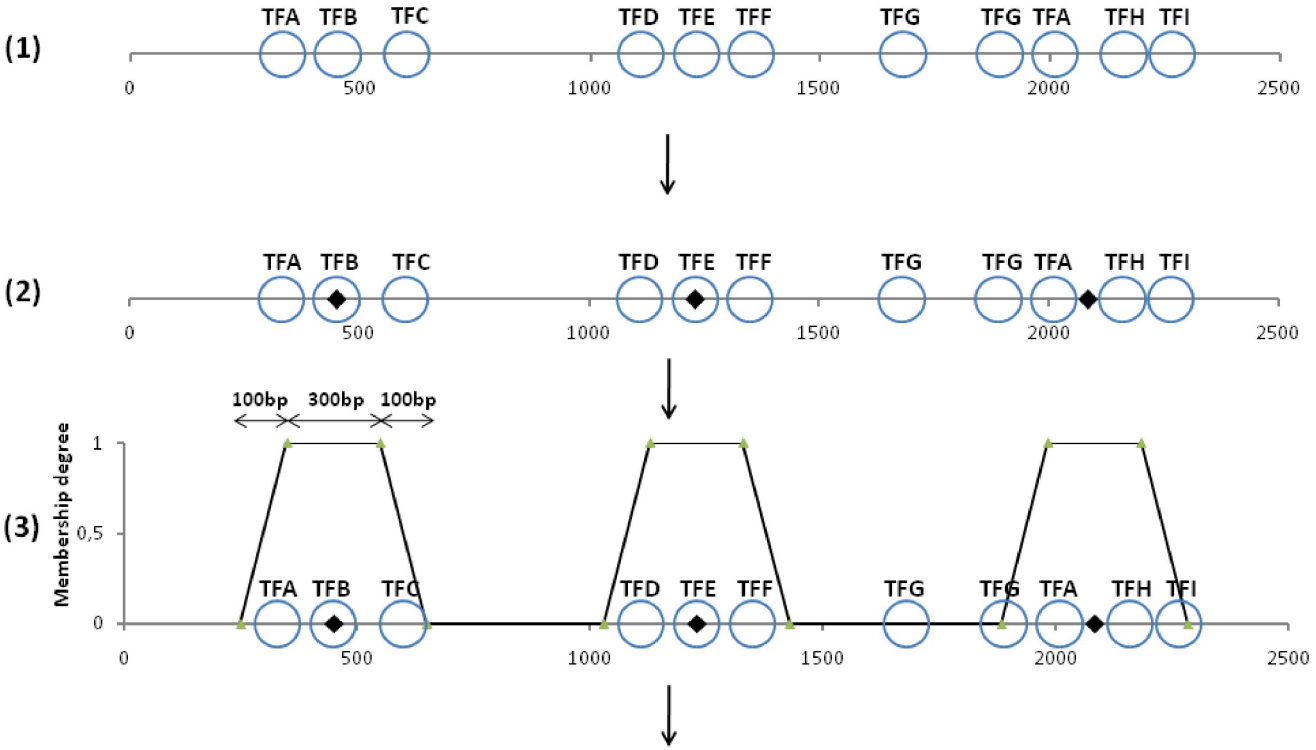
Post-processing the result set

The hierarchical clustering used in the first step of the proposed methodology yields a set of closely located TFBSs. However, especially in the case where a TFBS prediction tool has been used to predict a set of TFBSs to use as input, the effects of the presence of overlapping binding sites must be considered before generating the final result set. For instance, suppose the binding sites of the transcription in Figure 3. Previous approaches directly removed both binding sites in case of overlapping [39]. This action may lead to an incorrect counting of co-occurrence, since there could be a combination of binding sites which allows the simultaneous binding of both TFs (see Figure 3.a). Hence, we look for the optimum way of fitting a given TF combination (itemset) in a given fuzzy transaction, maximizing the membership degree of the itemset to the transaction (Figure 3.b). This optimum fit is considered a putative CRM.

Results and Discussion

We have developed an *in-silico* methodology to predict putative cis-regulatory modules (CRMs) which presents some interesting properties. First, using fuzzy sets to capture CRMs yields a more realistic model of these modules. In particular, softening their borders seems to fit the reality better than defining crisp partitions. In fact, fuzzy technology is proven to be a superior technology to enhance the interpretability of these partitions [40]. Moreover, the proposed fuzzy frequent itemset mining procedure allows to efficiently obtain any TF combination, overcoming constraints on the size and form of the recovered CRMs. Furthermore, when coupled with a TFBS prediction tool for scanning the entire non-coding genome, CisMiner does not limit the search to a set of specific regions.

The negative effects of the false positive locations generated by the TFBS search procedure are alleviated by the following filtering steps: i) Grouping the inferred TFBSs by means of a hierarchical clustering algorithm. The appearance of clusters of sites in small



Transaction ID	TFs (items)
1	TFA:1.0, TFB:1.0, TFC:1.0
2	TFD:1.0, TFE:1.0, TFF:1.0
3	TFG:0.01, TFA:1.0, TFH:1.0, TFI:0.23
...	...

Figure 2. Procedure for generating the fuzzy transactional database. (1) Each circle represents a binding site. Each binding site is labeled with the name of the TF which binds that BS. (2) Three clusters are obtained. Centroids are calculated for each cluster. (3) Fuzzy sets are defined for each cluster. (4) Fuzzy transactions are generated from the fuzzy sets. The value after the colon indicates the membership degree of the corresponding TF to the transaction.
doi:10.1371/journal.pone.0108065.g002

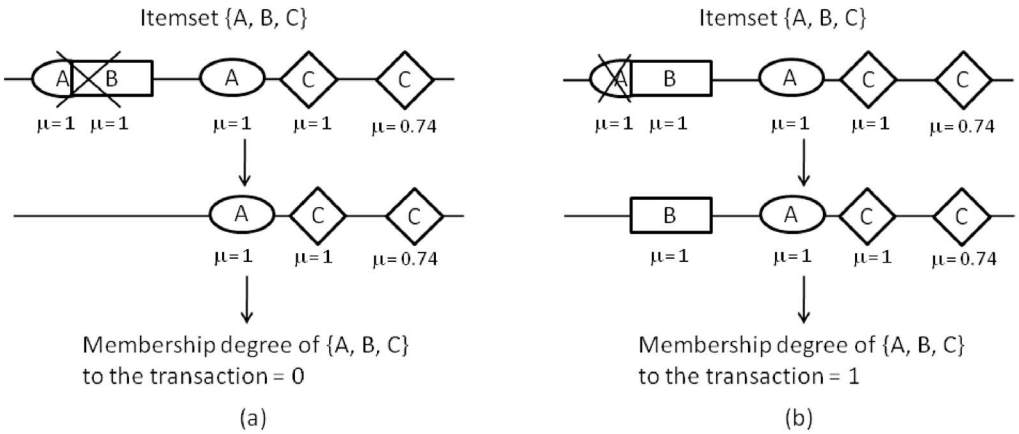


Figure 3. Post-processing of the results. The μ value indicates the membership degree of each binding site to its corresponding transaction. (a) Pairs of overlapping binding sites are directly removed. (b) The optimum way of fitting itemset {A, B, C} is found.
doi:10.1371/journal.pone.0108065.g003

regions of the genome is considered a reliable indicator of regulatory function [10, 39, 41, 42]. ii) Searching frequent itemsets among the obtained fuzzy groups. Frequent itemset mining procedures have been successfully applied in previous approaches [39, 43, 44]. Requiring the TF combinations to repeatedly appear will help to remove spurious occurrences. iii) Calculating the statistical significance of the obtained combinations. This provides a value indicating the reliability of the results, thus allowing to remove non-relevant combinations.

In order to test the performance of CisMiner, several experiments were made. First, we used our tool to perform CRM detection from a set of validated TFBSs. In this first experiment, we avoided the uncertainty associated with the computational inference of putative TFBSs and feed CisMiner with a set of validated TFBSs in *S. cerevisiae* to test if it is able to detect biologically significant combinations. Once the good performance of the methodology was proven given a set of validated TFBSs, we tested the performance of CisMiner when considering putative TFBSs inferred by a computational tool (Patser [45, 46]) in *S. cerevisiae* and *D. melanogaster*. Furthermore, a comparison to the equivalent crisp technique was made. Even though the advantages of using fuzzy techniques are intuitively clear, empirical data is provided to substantiate them.

CRM detection from validated TFBSs

This first approach was carried out over the yeast genome. More concretely, 3328 binding sites across the whole yeast genome were found for 102 TFs in the transcription factor binding site data published by Harbison et al. [47]. With these data, 570 transactions were obtained with a mean of 2.79 different TFs per transaction and a maximum of 10 TFs in a transaction [see File S2]. CisMiner obtained 36 itemsets from these transactions after setting the thresholds for the support and p-value to 0.01. The frequency of appearance of the 96 TFs which were included in at least one transaction is provided as File S1. These frequencies indicate that there are several not very-frequent TFs in the database which are likely to generate spurious itemsets.

In order to obtain evidence supporting the found relations, STRING [48] was used. Given a set of genes, STRING looks for associations among these genes at different levels: close location in the genome, co-occurrence of the queried genes across species, individual gene fusion events, co-expression, protein-protein interactions, curated databases and text mining. STRING provided evidence of relations among all the TFs present in the itemsets returned by CisMiner for 32 out of 36 obtained itemsets, representing an 88.8% of the results. Moreover, for 30 of the 36 itemsets, the graphs representing the associations among their TFs are *connected* graphs, i.e. there is a path connecting each pair of TFs in the graph. The other 2 of them contained indirect relations involving only one additional transcription factor. The complete set of graphs returned by STRING for this dataset is provided as File S1.

Table 1 shows the 20 most significant TF combinations according to the computed p-value. The whole list of the 36 obtained TF combinations can also be found in File S1. STRING returned a connected graph for all combinations in Table 1 except for itemsets 13 and 18. The relations among the TFs present in the predicted CRMs appear to be strong at several levels (see the corresponding graphs in supplementary material). Some of the CRMs suggested by CisMiner represent well characterised biological processes, such as the interaction of *SWI*6 with *MBP*1 and *SWI*4 (itemset 3) to form the MBF and SBF protein complexes, which cooperate and play a major role in progression from G1 to S phase of the cell cycle [49]. Indeed, it is worth mentioning that the combinations {*SWI*6,*MBP*1} and {*SWI*6,*SWI*4} are also present at positions 7 and 2, respectively. Another interesting relation is represented by itemsets 1, 5 and 14, regarding transcription factors *STE*12, *DIG*1, and *TEC*1. STRING returns a complete graph for these three TFs with strong empirical evindence, and a recent article by van der Felden et al. [50] also relates these three TFs. The co-ocurrence of *RAP*1 and *FHL*1 binding sites (itemset 9) is in concordance with previous results, since both of them were shown to bind upstream of many ribosomal protein genes [51].

No evidence was found by STRING for itemsets 13 and 18. It is noteworthy to state that when queried for the single TF *SUT*1, none of the genes connected to it in STRING appeared in the dataset by Harbison et al. Therefore, CisMiner was unable to find any STRING-proven relation of SUT1 to any other TF.

In addition to the STRING validation, we performed a further test using PubMed to get complementary literature-based evidences supporting the results. For the 36 obtained CRMs, 29 yield results when searched in PubMed, representing an 80.55% of the results. Itemsets 4, 13, 18 and 19 from table 1 did not yield any result when PubMed was queried. From these four itemsets, it is interesting to note that two (4 and 19) were nonetheless found as connected graphs in STRING.

Note that the lack of empirical evidence for some of the itemsets does not necessarily undermine the efectiveness of the method. Further studies and empirical evaluations are thus necessary to confirm the not-proven putative associations.

CRM detection from putative TFBSs in *S. cerevisiae*

CisMiner has been further tested with a set of putative computationally-predicted TFBSs in the *S. cerevisiae* genome. To this aim, data were obtained from the *Saccharomyces* Genome Database (SGD) [52, 53] and the JASPAR database [54]. In particular, the complete yeast genome was downloaded from the SGD. JASPAR provided 177 PWMs for the yeast genome (release of January 2014).

First, the locations of potential TFBSs were inferred. For this purpose, well-known Patser and Consite tools [46, 55] were used, since their good performance has been repeatedly proven [39, 56, 57]. Certainly, there exist more recent TFBS detection tools which take into account positional interdependencies within a putative TFBS sequence [58]. However, their applicability is constrained by the lack of available information; these techniques require to specify the lists of sequences used to calculate each PWM, which are not usually provided by the corresponding databases. In fact, JASPAR does not provide the sequences of any of the 177 yeast motifs it contains, while TRANSFAC [59] (public release 7.0 available online) does not provide sequences for any of the 24 yeast motifs stored.

We tested the ability of both Patser and Consite to recover real TFBSs using the yeast genome and the dataset by Harbison et al. as a benchmark. We first run Patser over the complete yeast genome. Given a PWM and a sequence, Patser yields a list of values (range [0,15]) indicating how well the PWM fits each position in the sequence. Thus, we needed to select a threshold to determine which of the positions were going to be considered putative TFBSs. In addition, some TFBSs may be easier to detect than others, since each motif presents its own peculiarities. Therefore, an independent Patser-score threshold for each motif was needed [60]. We realized that it was not feasible to set Patser-score thresholds under 7.5. This value was selected according to the Patser documentation [61] and to our own empirical experience. It was observed that setting the thresholds under 7.5 generates a huge number of putative TFBSs, thus diminishing the

Table 1. Top-20 TF combinations.

ID	Putative CRM	p-value	Supp.	Evidence
1	STE12, DIG1	1.11×10^{-16}	0.059	SP
2	SWI6, SWI4	1.11×10^{-16}	0.056	SP
3	SWI6, MBP1, SWI4	1.11×10^{-16}	0.025	SP
4	SKN7, SOK2, PHD1	3.00×10^{-15}	0.014	S
5	STE12, DIG1, TEC1	2.44×10^{-13}	0.018	SP
6	SOK2, PHD1	1.20×10^{-12}	0.027	SP
7	SWI6, MBP1	3.02×10^{-12}	0.043	SP
8	MBP1, SWI4	6.81×10^{-11}	0.038	SP
9	RAP1, FHL1	4.66×10^{-09}	0.016	SP
10	DIG1, SWI4, TEC1	2.02×10^{-08}	0.011	SP
11	DIG1, TEC1	4.74×10^{-08}	0.029	SP
12	AFT2, RCS1	4.83×10^{-08}	0.012	SP
13	PHD1, SUT1	5.92×10^{-08}	0.011	-
14	STE12, TEC1	7.19×10^{-08}	0.032	P
15	STE12, SWI6, SWI4	8.72×10^{-08}	0.014	SP
16	SWI6, DIG1, SWI4	2.13×10^{-07}	0.012	SP
17	FKH2, NDD1	3.23×10^{-07}	0.016	SP
18	SOK2, SUT1	8.78×10^{-07}	0.012	-
19	SKN7, SOK2	1.48×10^{-06}	0.022	S
20	SWI6, STB1	2.14×10^{-06}	0.012	SP

First dataset. The twenty TF combinations with the lowest p-value and highest support obtained when using the dataset by Harbison et al. Evidence column shows whether results were yielded when PubMed was queried for evidence in the literature (P), STRING [48] yielded a connected graph for the given TFs (S), both conditions (SP) or none (-) were met.

doi:10.1371/journal.pone.0108065.t001

significance of the results. Hence, for each motif, a specific Patser-score threshold over 7.5 was calculated. The selection of each threshold was done so that Patser was able to detect the maximum number of TFBSs described by Harbison et al. [47].

A similar procedure was carried out to test Consite's performance. In this case, for each PWM, Consite yields a list of values in the range [0,100]. Best results were obtained for a Consite threshold below 70. Figure 4 shows the number of *true* TFBSs recovered by each tool against the total number of sites detected. As it can be seen, Patser performs better than Consite in this particular case, since the total number of potential binding sites tends to be lower than that obtained with Consite for the same number of *true positives*. Therefore, Patser was finally selected for our purposes. The obtained thresholds for each motif are provided as File S1. Only 66 motifs are shown, since the rest of motifs were not found in the dataset by Harbison et al. We estimated the threshold for those PWMs not found in the dataset by Harbison by computing the median value of the thresholds obtained for the other 66 motifs [see File S1].

With these settings, 77921 putative binding sites were detected, which include 1412 of those described by Harbison et al. These 1412 TFBSs represent the $\sim$50% of the total number of binding sites described by these authors. It is noteworthy that, for a significant number of the motifs (e.g. ARR1, ASH1, BAS1), it was impossible to detect any of their known TFBSs, not even setting the Patser-score threshold to 0. Likewise, there were some motifs which required extremely low thresholds in order to capture their corresponding TFBSs (e.g 1.65, 2.25, 0.3 for ABF1, ADR1 & AFT2, respectively). All of this can be reflecting the biological complexity of the problem. It can also be be due to a certain incoherence between the data retrieved from JASPAR and those by Harbison et al. This could be due to the presence of outliers in both datasets or even to incoherences in the nomenclature of the motifs and TFs. The following tests are a way to show the ability of CisMiner to overcome these problems and yield some interesting results for further research from computationally inferred TFBSs. However, computational inference of TFBSs is out of the scope of this work.

Once the putative binding sites were detected, the fuzzy transactional database was built. 8176 transactions were obtained with an average of 8.67 different TFs per transaction and a maximum of 45 TFs in a transaction. The frequency of occurrence of each TF in the fuzzy transactional database is provided as File S1. The complete transactional data table is also provided in File S3.

CisMiner was run with the parameter settings summarized in Table 2. 255 itemsets were obtained. The reported combinations of TFs showed significant p-values, indicating that many of the obtained itemsets may represent real biological associations among the corresponding transcription factors. Moreover, the obtained itemsets contain between 2 and 4 TFs, which matches size estimations by previous works [42, 62]. Table 3 shows a sample of the obtained itemsets. The complete set of TF combinations is also provided as File S1. It is not the aim of this paper to provide a comprehensive list and biological interpretation of all of the obtained patterns, but to show that significant associations are obtained and that many of them are in concordance with

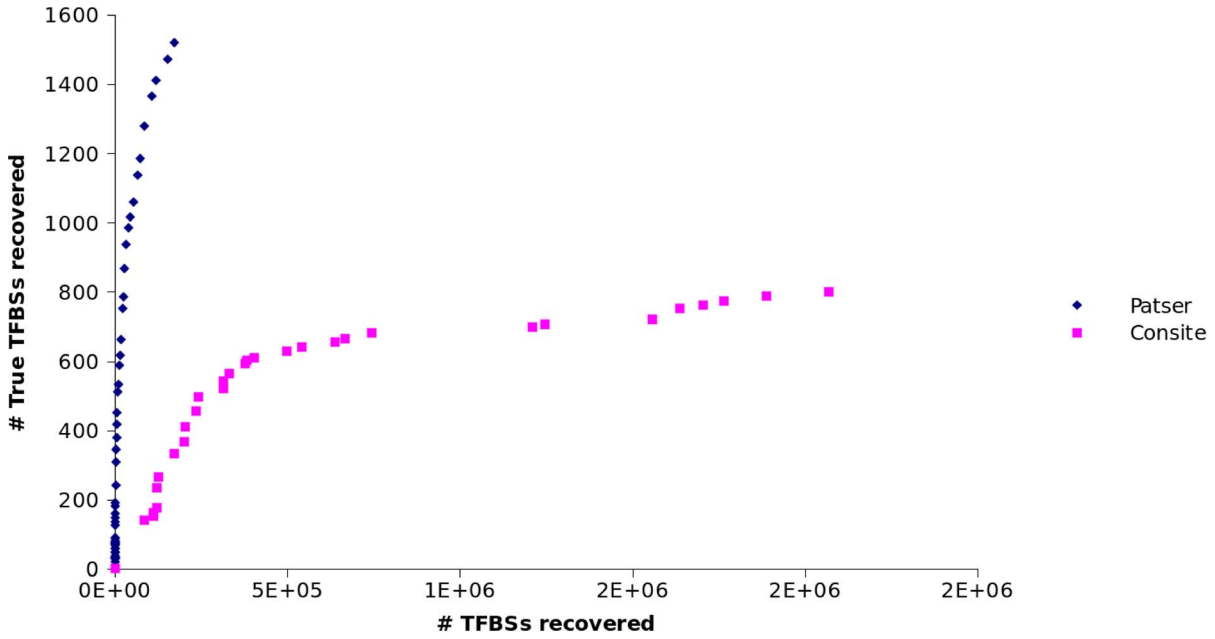


Figure 4. Comparison of Patser and Consite. Number of TFBSs from the dataset by Harbison et al. against the total number of TFBSs detected by Patser and Consite.
doi:10.1371/journal.pone.0108065.g004

previously published knowledge. A deeper biological analysis will be the topic for future works.

First, we wanted to check whether the methodology was able to detect the combinations obtained in Section following this second approach. Direct comparison of both result sets showed that only the itemset $\{STE12, TEC1\}$ was shared. The support threshold could be filtering out the rest of the itemsets in this second approach. In fact, lowering the support threshold we were able to get up to 13 of the previous combinations. Furthermore, the TFs DIG1, NDD1, SWI6, RCS1 and STB1 were not found in JASPAR, thereby justifying the absence of 14 of the itemsets that appeared in the first result set. Finally, the rest of itemsets lost their statistical significance.

Again, STRING was used to validate the results. In this case, STRING provided evidence of direct relations among the transcription factors of 23 of the recovered itemsets. In addition, for the first 100 results, STRING provided direct relations between the TFs of 11 of the putative CRMs, and indirect relations (involving only 1 additional TF) for 24. This implies that for the first 100 results, around 35% represent a direct or indirect relation among the proposed TFs. The itemset list is provided as File S1.

Here we briefly comment some of the results in Table 3. For example, STE12 and TEC1 seem strongly related (itemset 1). Both proteins are known to cooperate and regulate several cellular processes [63–65]. The ADR1 protein appears related to RAP1, RGT1 and SIP4 (itemsets 2–4). A number of bibliographic sources confirm such associations. ADR1 and RAP1, among other transcriptional regulators, may participate in barrier function, blocking the propagation of transcriptional silencing in yeast [66]. Likewise, the relation between ADR1 and RGT1 (YKL038W) was also found in the literature. These two factors are involved in the transcriptional response to transient perturbations in carbon source [67]. Finally, ADR1 and SIP4 participate in the transcriptional control of nonfermentative metabolism in the *Saccharomyces cerevisiae* [68].

Next itemset in Table 3 (itemset 5) contains AFT2 and RAP1, which are known to induce the expression of FRE1 in response to iron and copper depletion [53]. The next combination in Table 3 (itemset 6) involves the previously mentioned RGT1 factor, which in this case appears in cooperation with MIG3. Hazbun et al. [69] experimentally proved that both factors bind the promoter region of the gene SUC2, which product is an invertase enzyme. Then, three itemsets are shown which relate MSN4 with RPN4, SKN7

Table 2. Parameter values.

Patser score threshold	**Motifs with known BSs**	**Rest**
	See File S1	8.1
Hierarchical cluster	**Aggregation threshold**	
	300*bp*	
Fuzzy TD-FP-growth	**p-value threshold**	**Support threshold**
	0.01	0.01

Second dataset. Summary of the input parameters used.
doi:10.1371/journal.pone.0108065.t002

Table 3. TF combinations.

ID	Putative CRM	p-value	Support
1	STE12, TEC1	8.40×10^{-03}	0.013
2	ADR1, RAP1	2.39×10^{-08}	0.014
3	ADR1, RGT1	9.18×10^{-05}	0.013
4	ADR1, SIP4	1.22×10^{-04}	0.011
5	AFT2, RAP1	1.17×10^{-13}	0.012
6	MIG3, RGT1	3.13×10^{-06}	0.012
7	MSN4, RPN4	2.37×10^{-03}	0.024
8	MSN4, SKN7	9.24×10^{-06}	0.013
9	MSN4, GIS1	1.11×10^{-16}	0.020
10	MIG1, MIG2	1.11×10^{-16}	0.016
11	STE12, GCR2, STB5, XBP1	1.11×10^{-16}	0.010
12	ADR1, MIG1	1.11×10^{-16}	0.020
13	SUT1, MIG1	1.11×10^{-16}	0.018

Second dataset. Some of the TF combinations obtained when using the TFBSs detected by Patser (yeast genome).
doi:10.1371/journal.pone.0108065.t003

and GIS1 (itemsets 7–9), all of them representing previously described associations. Thus, MSN4, RPN4 and SKN7 are known to participate in the transcriptional response of *Saccharomyces cerevisiae* to the stress imposed by certain fungicides and herbicides [70, 71]. Regarding the relation between MSN4 and GIS1, STRING returned associations at different levels: experimental, curated databases and text mining. Previous authors described that the Rim15 regulon is mediated by these two transcription factors [72]. Finally, MIG1 and MIG2 also appear strongly related at different levels. Glucose repression of the SUC2 gene is dependent on MIG1 and MIG2 [73]. The last itemsets in Table 3 show some combinations for which STRING returned indirect associations, and some others for which no confirmation was obtained.

As in the previous section, a search over PubMed was performed in order to obtain more information about the complete result set. The PubMed results were manually curated and at least 23 more combinations were found to be related [see File S1].

In our aim to provide additional experimentation supporting the obtained results, the full methodology was re-run over 100 randomized yeast genomes. Interestingly, the mean number of predicted TFBSs by Patser in this random datasets was 73307.9, which is near the 77921 TFBSs predicted using the real dataset. Even more interestingly, only an average of 11.5 significant TF combinations remained in this case, in constrast to the 255 putative CRMs obtained using the real genome. This fact could be a clear evidence of the potential of the methodology to remove spurious occurrences, and suggests a false discovery rate lower than 5%.

To finish this section, it is worth mentioning that the complete process was also re-run after raising the minimum Patser-score threshold to 8. Thus, 56 itemsets were obtained (results not shown). It is interesting that this new set of combinations was a subset of the 255 itemsets analyzed in this section. Moreover, the *p*-values were almost the same to those calculated when setting the minimum Patser-score threshold to 7.5. This fact suggests that the methodology is coherent and robust.

CRM detection from putative TFBSs in *D. melanogaster*

In order to test whether the application of the methodology over larger and more complex genomes is computationally feasible and whether it can still unveil significant knowledge, a similar experiment was performed over the *Drosophila melanogaster* genome.

A comprehensive biological interpretation of the complete result set was out of the scope of this work. However, the interest of the results is discussed and verified in this section. The complete *Drosophila melanogaster* genome was downloaded from FlyBase [74, 75] (Release 5.56, March 2014). In addition, 131 PWMs of the *Drosophila melanogaster* genome were retrieved from Jaspar. According to the Patser-score thresholds calculated above, subsequences scoring over 8.1 were considered as potential TFBSs. Thus, 152338 transactions were obtained with a mean of 4.34 different TFs per transaction and a maximum of 15 TFs in a transaction. The frequency of appearance of each TF in the fuzzy transactional database is provided as File S1. The complete transactional data table is provided as File S4.

Interestingly, only 39 significant putative CRMs were obtained. In this case, STRING returned just one direct relation ($\{Mad,brk\}$, Table 4). Other 8 itemsets returned indirect relations when STRING was queried, representing $\sim$20% of the total obtained. Regarding PubMed searches, 12 of the results returned matches when searched in PubMed. The complete list with the corresponding PubMed links is provided in File S1. However, it is extremely difficult to search for scientific evidences of the obtained relations due to the unspecific identifiers of these transcription factors, e.g.: *opa*, *btd*, *h*, *D*. The putative CRMs suggested by at least two more itemsets were found to be related by a manual inspection of the literature retrieved by PubMed. This last experiment has shown that it is computationally feasible to apply the methodology over more complex genomes. Moreover, these 39 modules present significant quality values and may represent real biological interactions. Future work is needed to biologically validate and interpret all the obtained combinations.

Fuzzy-crisp comparison

To evaluate the contribution of fuzzy technologies to the methodology, we implemented a crisp version of the same methodology and compared the results obtained using the same quality thresholds (See Methods). The procedure to create the crisp transactional database was a crisp version of the fuzzy procedure previously described. First, a crisp clustering method was run to identify the transactions, where crisp borders were defined for each cluster at bases $C-300$ and $C+300$ (C is the cluster centroid as defined in Methods). In the crisp transactional database, all items in each transaction have a membership degree of 1.0. This means that a TFBS belongs to a given cluster if it is fully located within bases $C-300$ and $C+300$. A crisp version of the frequent itemset mining algorithm was applied in this case to extract the frequent itemsets.

As expected, significant differences were found between the crisp and fuzzy results on the yeast dataset. First, the crisp algorithm obtained 4217 combinations while the fuzzy one returned 1388, being these a subset of the crisp results [see File S5]. In order to determine whether the values of the measures obtained by the fuzzy and the crisp methodologies were significantly different, two ANOVAs were carried out (Table 5). Fuzzy sets are proven to be a superior technology to model partitions with blurry borders. Obtained results show more significant p-values achieved by the fuzzy methodology, suggesting that an effective prunning of spurious itemsets is achieved by applying the fuzzy algorithm.

The same steps were taken to test fuzzy-crisp performance over the *Drosophila melanogaster* dataset. In this case, 4388 TF combinations were returned by the crisp procedure while 4272 were obtained by the fuzzy one [see File S5]. Table 5 shows that the results obtained with this other dataset also comply with the previous comments. These results show the advantages of using fuzzy technology to model the TFBS clusters, improving the representation of a CRM by removing sharp borders and achieving a better performance in terms of p-values.

Conclusions

In this work we have presented CisMiner, an *in silico* fuzzy methodology able to obtain putative CRMs by means of extracting significant co-occurrences of closely located TFBSs genome-wide. This methodology presents some interesting properties:

Genome-wide scope. CisMiner is capable of analyzing the complete non-coding genome of an organism without limiting the search space to specific regions.

Uncertainty handling. The uncertainty inherent to CRM detection is better modeled by fuzzy technology.

Flexibility in TFBS number, distribution and distance. CisMiner does not impose constraints on the form or number of elements of the discovered CRMs.

Prior knowledge not needed. The proposed methodology does not require any prior knowledge to restrict the search space.

Easily interpretable results. Each CRM is expressed as a set of TFs, its frequency of appearance in the genome and a p-value, which makes the results robust and easily interpretable.

Extensible and efficient. To the extent of our knowledge, the proposed methodology is the only one available that addresses a global genome-wide search for CRMs with no restrictions, which is efficient and robust due to the use of efficient data structures. In addition, it is easily extensible to larger genomes.

Freely accessible. CisMiner is available at: http://genome2.ugr.es/cisminer.

Several experiments were carried out to validate the proposed methodology. CisMiner has been shown to identify statistically significant CRMs composed of TFs with well-known interactions as reported in STRING and the literature.

The experimental results also showed that when coupled with a TFBS detection tool, the performance of the final results is strongly dependent on the performance of the TFBS detection approach.

On the other hand, although many approaches have been proposed to understand local regulatory mechanisms, little has been proposed to extract knowledge about broader regulatory mechanisms. In this sense, the genome-wide scope of CisMiner can help to shed some light in such regulatory processes.

Future work comprises testing the methodology on more complex species with larger genomes. The performed experiments have shown that the methodology is based on consistent principles and its modularity enables us to easily improve it when new methos and data are made available. In addition, we believe that the integration of additional sources of information besides sequence data (e.g. chromatin structure, protein structure) may help to refine the results.

Table 4. TF combinations.

Id	Putative CRM	p-value	Support
1	btd, hkb	1.11×10^{-16}	0.045
2	btd, Mad	1.11×10^{-16}	0.041
3	btd, opa	1.11×10^{-16}	0.033
4	btd, h	1.11×10^{-16}	0.003
5	Mad, brk	1.11×10^{-16}	0.029
6	btd, CTCF	1.11×10^{-16}	0.023
7	Mad, opa	1.11×10^{-16}	0.023
8	Mad, hkb	1.11×10^{-16}	0.022
9	brk, opa	1.11×10^{-16}	0.021
10	Mad, h	1.11×10^{-16}	0.019

Third dataset. Some of the TF combinations obtained when using the TFBSs detected by Patser (Drosophila genome).
doi:10.1371/journal.pone.0108065.t004

Table 5. Fuzzy-crisp comparison.

	Yeast	Dmel
Mean fuzzy support	0.0084	0.0088
Mean crisp support	0.0108	0.0094
Mean fuzzy *p*-value	0.0144	0.0133
Mean crisp *p*-value	0.0070	0.0104
ANOVA support significance	$<10^{-38}$	$<10^{-8}$
ANOVA *p*-value significance	$<10^{-19}$	$<10^{-8}$

The four first rows show the mean values of fuzzy/crisp support and *p*-value of the combinations respectively. The last two rows show the statistical significance returned by the ANOVA procedure.
doi:10.1371/journal.pone.0108065.t005

Supporting Information

File S1 Complete datasets. Additional pdf file including the full datasets obtained for both *Drosophila* and *saccharomyces*, along with the selected thresholds and STRING graphs for the first dataset.
(PDF)

File S2 Fuzzy transactions. First dataset. Fuzzy transactional database built when considering the dataset of real TFBSs reported by Harbison et al.
(TXT)

File S3 Fuzzy transactions. Second dataset. Fuzzy transactional database built when considering the TFBSs detected by Patser (*S. cerevisiae* genome).
(TXT)

File S4 Fuzzy transactions. Third dataset. Fuzzy transactional database built when considering the TFBSs detected by Patser (*D. melanogaster* genome).
(ZIP)

File S5 Fuzzy and crisp comparison. Three different tables containing the set of fuzzy itemsets used for the comparison, the set of crisp itemsets and a list with the intersection between both result sets.
(ZIP)

Author Contributions

Conceived and designed the experiments: AB FJL FG CC. Performed the experiments: CN FJL. Analyzed the data: CN FJL. Contributed reagents/materials/analysis tools: FJL FG CC CN. Wrote the paper: CN AB FG CC. Website implementation: CN.

References

1. Alberts B, Johnson A, Lewis J, Raff M, Roberts K, et al. (2002) Molecular Biology of The Cell. New York, NY, USA: Garland Science.
2. Spitz F, Furlong EE (2012) Transcription factors: from enhancer binding to developmental control. Nature Reviews Genetics 13: 613–626.
3. Sun H, Guns T, Fierro AC, Thorrez L, Nijssen S, et al. (2012) Unveiling combinatorial regulation through the combination of chip information and in silico cis-regulatory module detection. Nucleic acids research 40: e90–e90.
4. van Loo P, Marynen P (2009) Computational methods for the detection of cis-regulatory modules. Briefings in Bioinformatics 10(5): 509–524.
5. Frith MC, Hansen U, Weng Z (2001) Detection of cis-element clusters in higher eukaryotic dna. Bioinformatics 17: 878–889.
6. Frith MC, Li MC, Weng Z (2003) Cluster-buster: Finding dense clusters of motifs in dna sequences. Nucleic acids research 31: 3666–3668.
7. Sinha S, Van Nimwegen E, Siggia ED (2003) A probabilistic method to detect regulatory modules. Bioinformatics 19: i292–i301.
8. Herrmann C, Van de Sande B, Potier D, Aerts S (2012) i-cistarget: an integrative genomics method for the prediction of regulatory features and cis-regulatory modules. Nucleic acids research 40: e114–e114.
9. Nandi S, Blais A, Ioshikhes I (2013) Identification of cis-regulatory modules in promoters of human genes exploiting mutual positioning of transcription factors. Nucleic acids research 41: 8822–8841.
10. Vavouri T, Elgar G (2005) Prediction of cis-regulatory elements using binding site matrices-the successes, the failures and the reasons for both. Current opinion in genetics & development 15(4): 395–402.
11. Blanchette M, Bataille AR, Chen X, Poitras C, Laganière J, et al. (2006) Genome-wide computational prediction of transcriptional regulatory modules reveals new insights into human gene expression. Genome research 16: 656–668.
12. Hallikas O, Palin K, Sinjushina N, Rautiainen R, Partanen J, et al. (2006) Genome-wide prediction of mammalian enhancers based on analysis of transcription-factor binding affinity. Cell 124: 47–59.
13. Ren B, Robert F, Wyrick JJ, Aparicio O, Jennings E, et al. (2000) Genome-wide location and function of DNA binding proteins. Science's STKE 290(5500): 2306–2309.
14. Cawley S, Bekiranov S, Ng HH, Kapranov P, Sekinger EA, et al. (2004) Unbiased mapping of transcription factor binding sites along human chromosomes 21 and 22 points to widespread regulation of noncoding rnas. Cell 116(4): 499–509.
15. Impey S, McCorkle SR, Cha-Molstad H, Dwyer JM, Yochum GS, et al. (2004) Defining the CREB Regulon A Genome-Wide Analysis of Transcription Factor Regulatory Regions. Cell 119(7): 1041–1054.
16. Matyash A, Chung HR, Jäckle H (2004) Genome-wide mapping of in vivo targets of the Drosophila transcription factor Krüppel. Journal of Biological Chemistry 279(29): 30689–30696.
17. Testa A, Donati G, Yan P, Romani F, Huang THM, et al. (2005) Chromatin Immunoprecipitation(ChIP) on Chip Experiments Uncover a Widespread Distribution of NF-Y Binding CCAAT Sites Outside of Core Promoters. Journal of Biological Chemistry 280(14): 13606–13615.
18. Lettice LA, Heaney SJH, Purdie LA, Li L, de Beer P, et al. (2003) A long-range Shh enhancer regulates expression in the developing limb and fin and is associated with preaxial polydactyly. Human Molecular Genetics 12(14): 1725–1735.
19. Nobrega MA, Ovcharenko I, Afzal V, Rubin EM (2003) Scanning human gene deserts for long-range enhancers. Science 302(5644): 413.
20. Woolfe A, Goodson M, Goode DK, Snell P, McEwen GK, et al. (2005) Highly conserved non-coding sequences are associated with vertebrate development. PLoS Biol 3(1): e7.
21. Dermitzakis ET, Clark AG (2002) Evolution of transcription factor binding sites in Mammalian gene regulatory regions: conservation and turnover. Molecular Biology and Evolution 19(7): 1114–1121.
22. Costas J, Casares F, Vieira J (2003) Turnover of binding sites for transcription factors involved in early Drosophila development. Gene 310: 215–220.
23. Emberly E, Rajewsky N, Siggia ED (2003) Conservation of regulatory elements between two species of Drosophila. BMC bioinformatics 4(1): 57–67.
24. Rohr CO, Parra RG, Yankilevich P, Perez-Castro C (2013) Insect: In-silico search for co-occurring transcription factors. Bioinformatics 29: 2852–2858.
25. Nikulova AA, Favorov AV, Sutormin RA, Makeev VJ, Mironov AA (2012) Coreclust: identification of the conserved crm grammar together with prediction of gene regulation. Nucleic acids research 40: e93–e93.
26. Laimer J, Zuzan CJ, Ehrenberger T, Freudenberger M, Gschwandtner S, et al. (2013) D-light on promoters: a client-server system for the analysis and visualization of cis-regulatory elements. BMC bioinformatics 14: 140.
27. Ha N, Polychronidou M, Lohmann I (2012) Cops: detecting co-occurrence and spatial arrangement of transcription factor binding motifs in genome-wide datasets. PloS one 7: e52055.
28. Jegga AG, Sherwood SP, Carman JW, Pinski AT, Phillips JL, et al. (2002) Detection and visualization of compositionally similar cis-regulatory element

clusters in orthologous and coordinately controlled genes. Genome research 12: 1408–1417.
29. Deyneko IV, Kel AE, Kel-Margoulis OV, Deineko EV, Wingender E, et al. (2013) Matrixcatch-a novel tool for the recognition of composite regulatory elements in promoters. BMC bioinformatics 14: 1–10.
30. Leoncini M, Montangero M, Pellegrini M, Tillán KP (2013) Cmf: a combinatorial tool to find composite motifs. In: Learning and Intelligent Optimization, Berlin Heidelberg: Springer. pp. 196–208.
31. Lopez FJ, Blanco A, Garcia F, Cano C, Marin A (2008) Fuzzy association rules for biological data analysis: a case study on yeast. BMC Bioinformatics 9: 107–115.
32. Zadeh LA (1965) Fuzzy sets. Information and Control 8(3): 338–353.
33. Agrawal R, Imielinski T, Swami A (1993) Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD INTL Conf. on Management of Data (ACM SIGMOD 93); Washington, USA. pp. 207–216.
34. Ceglar A, Roddick JF (2006) Association mining. ACM Computing Surveys 38(2), Article 5: 1–42.
35. Naulaerts S, Meysman P, Bittremieux W, Vu TN, Berghe WV, et al. (2013) A primer to frequent itemset mining for bioinformatics. Briefings in bioinformatics: bbt074.
36. Arnone MI, Davidson EH (1997) The hardwriting of development: organization and function of genomic regulatory systems. Development 124(10): 1851–1864.
37. Delgado M, Marin N, Sanchez D, Vila MA (2003) Fuzzy association rules: General model and applications. IEEE Trans Fuzzy Systems 11: 214–225.
38. Gallo A, de Bie T, Cristianini N (2007) MINI: Mining informative non-redundant itemsets. Lecture Notes in Computer Science 4702: 438–445.
39. Morgan XC, Ni S, Miranker DP, Iyer VR (2007) Predicting combinatorial binding of transcription factors to regulatory elements in the human genome by association rule mining. BMC Bioinformatics 8: 445–458.
40. Delgado M, Marin N, Martin-Bautista MJ, Sanchez D, Vila MA (2003) Mining fuzzy association rules: an overview. In: Proceedings of the BISC International Workshop on Soft Computing for Internet and Bioinformatics; Berkeley, CA, USA.
41. Frith MC, Hansen U, Weng Z (2001) Detection of cis-element clusters in higher eukaryotic DNA. Bioinformatics 17(10): 878–889.
42. Blanchette M, Bataille AR, Chen X, Poitras C, Laganière J, et al. (2006) Genome-wide computational prediction of transcriptional regulatory modules reveals new insights into human gene expression. Genome Research 16(5): 656–668.
43. Sun H, de Bie T, Storms V, Fu Q, Dhollander T, et al. (2009) ModuleDigger: an itemset mining framework for the detection of cis-regulatory modules. BMC bioinformatics 10(Suppl 1): S30.
44. Pham TH, Satou K, Ho TB (2004) Mining yeast transcriptional regulatory modules from factor DNA-binding sites and gene expression data. Genome Informatics Series 15(2): 287–295.
45. Hertz GZ, Hartzell III GW, Stormo GD (1990) Identification of consensus patterns in unaligned DNA sequences known to be functionally related. Bioinformatics 6(2): 81–92.
46. Hertz GZ, Stormo GD (1999) Identifying dna and protein patterns with statistically significant alignments of multiple sequences. Bioinformatics 15: 563–577.
47. Harbison CT, Gordon DB, Lee TI, Rinaldi NJ, Macisaac KD, et al. (2004) Transcriptional regulatory code of a eukaryotic genome. Nature 431(7004): 99–104.
48. Franceschini A, Szklarczyk D, Frankild S, Kuhn M, Simonovic M, et al. (2013) String v9. 1: protein-protein interaction networks, with increased coverage and integration. Nucleic acids research 41: D808–D815.
49. Koch C, Moll T, Neuberg M, Ahorn H, Nasmyth K (1993) A role for the transcription factors Mbp1 and Swi4 in progression from G1 to S phase. Science 261(5128): 1551–1557.
50. van der Felden J, Weisser S, Brückner S, Lenz P, Mösch HU (2014) The transcription factors tec1 and ste12 interact with co-regulators msa1 and msa2 to activate adhesion and multicellular development. Molecular and cellular biology: MCB–01599.
51. Schawalder SB, Kabani M, Howald I, Choudhury U, Werner M, et al. (2004) Growth-regulated recruitment of the essential yeast ribosomal protein gene activator Ifh1. Nature 432(7020): 1058–1061.
52. Cherry JM, Hong EL, Amundsen C, Balakrishnan R, Binkley G, et al. (2012) Saccharomyces genome database: the genomics resource of budding yeast. Nucleic acids research 40: D700–D705.
53. The saccharomyces genome database. Available: http://www.yeastgenome.org.
54. Mathelier A, Zhao X, Zhang AW, Parcy F, Worsley-Hunt R, et al. (2014) Jaspar 2014: an extensively expanded and updated open-access database of transcription factor binding profiles. Nucleic acids research 42: D142–D147.
55. Sandelin A, Wasserman WW, Lenhard B (2004) ConSite: web-based prediction of regulatory elements using cross-species comparison. Nucleic Acids Research 32(Web Server Issue): W249.
56. Salgado H, Gama-Castro S, Martinez-Antonio A, Diaz-Peredo E, Sanchez-Solano F, et al. (2004) RegulonDB (version 4.0): transcriptional regulation, operon organization and growth conditions in Escherichia coli K-12. Nucleic Acids Research 32(Database Issue): D303.
57. Jones EA, Flavell RA (2005) Distal enhancer elements transcribe intergenic RNA in the IL-10 family gene cluster. The Journal of Immunology 175(11): 7437–7446.
58. Tomovic A, Oakeley EJ (2007) Position dependencies in transcription factor binding sites. Bioinformatics 23(8): 933–941.
59. Wingender E, Chen X, Hehl R, Karas H, Liebich I, et al. (2000) TRANSFAC: an integrated system for gene expression regulation. Nucleic Acids Research 28(1): 316–319.
60. Turatsinze JV, Thomas-Chollier M, Defrance M, van Helden J (2008) Using RSAT to scan genome sequences for transcription factor binding sites and cis-regulatory modules. Nat Protoc 3(10): 1578–1588.
61. Rsa-tools-patser. Available: http://rsat.ulb.ac.be/rsat/patser_form.cgi.
62. Ryu T, Kim Y, Kim DW, Lee D (2007) Computational identification of combinatorial regulation and transcription factor binding sites. Biotechnology and Bioengineering 97(6): 1594–1601.
63. Chou S, Lane S, Liu H (2006) Regulation of mating and filamentation genes by two distinct Ste12 complexes in Saccharomyces cerevisiae. Molecular and Cellular Biology 26(13): 4794–4805.
64. Lo WS, Dranginis AM (1998) The cell surface flocculin Flo11 is required for pseudohyphae formation and invasion by Saccharomyces cerevisiae. Molecular Biology of the Cell 9(1): 161–171.
65. Kim TS, Lee SB, Kang HS (2004) Glucose repression of STA1 expression is mediated by the Nrg1 and Sfl1 repressors and the Srb8-11 complex. Molecular and Cellular Biology 24(17): 7695–7706.
66. Yu Q, Qiu R, Foland TB, Griesen D, Galloway CS, et al. (2003) Rap1p and other transcriptional regulators can function in defining distinct domains of gene expression. Nucleic Acids Research 31(4): 1224–1233.
67. Ronen M, Botstein D (2006) Transcriptional response of steady-state yeast cultures to transient perturbations in carbon source. Proceedings of the National Academy of Sciences of the United States of America 103(2): 389–394.
68. Schüller HJ (2003) Transcriptional control of nonfermentative metabolism in the yeast Saccharomyces cerevisiae. Current genetics 43(3): 139–160.
69. Hazbun TR, Fields S (2002) A genome-wide screen for site-specific DNA-binding proteins. Molecular & Cellular Proteomics 1(7): 538–543.
70. Santos PM, Simões T, Sá-Correia I (2009) Insights into yeast adaptive response to the agricultural fungicide mancozeb: A toxicoproteomics approach. Proteomics 9(3): 657–670.
71. Teixeira MC, Fernandes AR, Mira NP, Becker JD, Sá-Correia I (2006) Early transcriptional response of Saccharomyces cerevisiae to stress imposed by the herbicide 2, 4-dichlorophenoxyacetic acid. FEMS yeast research 6(2): 230–248.
72. Cameroni E, Hulo N, Roosen J, Winderickx J, de Virgilio C (2004) The novel yeast PAS kinase Rim 15 orchestrates G0-associated antioxidant defense mechanisms. Cell Cycle 3(4): 462–468.
73. Lutfiyya LL, Iyer VR, de Risi J, de Vit MJ, Brown PO, et al. (1998) Characterization of three related glucose repressors and genes they regulate in Saccharomyces cerevisiae. Genetics 150(4): 1377–1391.
74. Tweedie S, Ashburner M, Falls K, Leyland P, McQuilton P, et al. (2009) Flybase: enhancing drosophila gene ontology annotations. Nucleic acids research 37: D555–D559.
75. Flybase: A database of drosophila genes & genomes. Available: http://flybase.org.