BMA – Boolean Matrices as Model for Motif Kernels

Jan Schröder, Manfred Schimmler, Heiko Schröder, Karsten Tischer

Christian Albrechts Universität, Institut für Informatik, Kiel, Germany
{jasc|masch}@informatik.uni-kiel.de
RMIT University, NICTA VRL, Melbourne, Australia
heiko@cs.rmit.edu.au
Freie Universität, Institut für Virologie, Berlin, Germany
k.tischer@fu-berlin.de

Abstract

We introduce the data model BM, which specifies kernels of motifs by means of Boolean matrices. Different from position frequency matrices these only specify which bases can appear in which position of a motif instance. Boolean matrices describe motifs still more precisely than models based on consensus strings and Hamming distance and thus allow keeping the number of false positives low.

Based on the BM model we introduce a new algorithm BMA for motif discovery that attempts to reduce the number of false positives as much as possible. The main idea is to start with small kernels of motif instances and iteratively enlarge the kernels with sets of strings which have a high expected signal to noise ratio, thus keeping the signal to noise ratio as high as possible. Only after a kernel of substantial size has been found, we use position frequency matrices and add in the final stage of the algorithm BMA strings that are close to the kernel to form the final list of potential motif instances.

The data model BM simplifies theoretical analysis. For this we restrict ourselves further to Boolean matrices which have either one or two "1"s in each column, i.e. the positions are either preserved or semi-preserved. Finally we compare BMA to other word based approaches on synthetic data fitting the BM model.

1 Introduction

1.1 Background

In [4] the algorithm IGOM has been presented, which is based on the SP-model (also allowing only preserved and semi-preserved positions in a model of a motif). While the performance of IGOM came close to the performance of BMA, BMA has the advantage of being simpler and being able to be expanded easily to cases where more than two different bases can appear in a position of a motif instance. BMA is also closer to what seems intuitively right, e.g. if all motif instances have a G in position *i* and an A or a T in position *j*, then a C in position *i* is met with the same penalty as a C in position *j*. This is very different from the way PWMs are typically applied.

For a detailed overview on the topic of finding motifs in DNA sequences we also refer to [4] where the problem itself and the scientific approaches to its solution are discussed. The reader might consider [4] as well to fully understand some of the ideas developed in this article because the BMA is a enhancement of the IGOM-algorithm.

In this paper we give a close comparison to the projection algorithm and demonstrate on synthetic data that BMA significantly outperforms the projection algorithm.

1.2 Major features of BMA

Theoretical analysis of the algorithms we base on modelling motifs by means of Boolean matrices in which all positions within a motif are either preserved or semi-preserved, i.e. the matrix has either one or two "1"s in each column. The main characteristics of the BMA algorithm are:

1. It is based on Boolean matrices, which provide a relatively precise description of sets of motif instances.
2. It builds up motif kernels iteratively starting with small motif kernels (typically of size 1).
3. In each of its steps it keeps the signal to noise ratio high and thus minimizes the number of false positives.

Applying BMA to a dataset from Staphylococus Aureus [3] has shown that we were able to detect a range of previously unnoticed motif instances – a case where popular motif finding methods had failed to detect these.

1.3 Content

In Chapter 2 we present basic notation and techniques used in BMA. In Chapter 3 we present BMA. Chapter 4 demonstrates that BMA maximizes the signal to noise ratio. Chapter 5 presents some simulation results that demonstrate that BMA outperforms the projection algorithm and finally Chapter 6 discusses further research needed in this area.

2. Notation and elementary techniques

Throughout this paper we use *l* for the motif length, *m* for the number of motif instances present in the input data and *n* for the size of input data. The algorithm we present iteratively generates sets of potential motif instances, starting from sets of strings of size one. We call sets of potential motif instances *motif kernels*.

The projection algorithm is based on the assumption that motifs (or sets of motif instances) are appropriately described using a consensus string and a maximal Hamming distance. But the Hamming distance is not a good metric for measuring distance between motifs, which can easily be understood by looking at *position frequency matrices* (PFM) [2]. They describe better (with more flexibility and also more precision), whether a string is likely to be an instance of a motif or not. Thus within BMA we model motifs using PFMs and in the first stages of BMA we use substantially simplified versions of PFM, i.e. we use Boolean matrices instead.

In the final stage of BMA we use the PFM to score strings (in order to judge, whether they should be considered as motif instances or not). Let $X=(x_1, x_2, \ldots, x_l)$, be a string. Let PFM be a *l* by *4* matrix PFM(i,j) with $1 \le i \le l$ and *j* in {A, C, G, T}. The score of *X* is defined as $score(PFM, X) := \sum_{i=1}^{l} PFM(i, x_i)$. We also need the maximal possible score for a PFM, which is the sum of the maxima of all its columns.

Several researchers have converted Position Frequency Matrices into Position Weight Matrices (PWM) for scoring purposes [2]. For our purpose PFMs seem to be sufficient and it would only be a minor and probably insignificant change to use PWMs within our approach. We intend to include an examination of the use of different types of PWMs in a later project in order to substantiate the above claim.

2.1. The data model BM

Throughout this paper we assume that the Boolean matrices that describe motif kernels have either one or two "1"s in each column. This could be relaxed, but it would result in changes to the corresponding algorithm BMA. We use p for the number of preserved positions (those that have only one "1") and sp for the number of semi-preserved positions (those that have two "1"s), p+sp=*l*. Within BM the performance of motif finding algorithms can in some cases be analyzed theoretically and it is particularly easy to calculate signal to noise ratios within this model.

Let the number of "0"s in column *i* of a Boolean matrix be Z_i and the number of "1"s be N_i then the number of different strings that have exactly one mismatch with the matrix is given by $\Sigma_i (Z_i * \prod_{j \neq i} N_j)$. This makes it very simple to calculate the expected number of false positives within the BM model (see Section 2.2).

The following notation is used in order to describe BMA.

Notation:

Let K be any set of strings of length *l*.

1) K defines a *4xl* Boolean matrix BM_K such that $BM_K(i,b)=1$ if and only if there is a string X in K with X(i)=b, (0 otherwise).
2) A string X has exactly one mismatch with a Boolean matrix BM if and only if there is exactly one position i with $BM(i, X_i)=0$ ($BM(j,X_j)=1$ for $j \neq i$)
3) Whenever we search for strings in the input data, we regard two identical strings in different positions of the input data as two strings (i.e. they are distinguished by their positions).
4) $D1_K$:= set of all strings of length l of the input data, which have exactly one mismatch with BMK.
5) We use the term "applying a BM" for finding all *l*-mers within the input data, that match the BM.

2.2. Signal to noise ratio in motif discovery algorithms

Signal to noise ratio is an important term to describe the quality of acoustical signals, where (often due to transmission problems) the desired data, is "polluted" with noise and thus not recognisable anymore. Motif discovery algorithms deal with sets of strings of which we hope the majority to be motif instances, but even

algorithms of highest quality are likely to also return strings that match the given search criteria, but actually are not motif instances. Such strings we call false positives, they correspond to the noise in acoustical signals. Different from acoustics, in motif finding and motif discovery a signal to noise ratio of 1 might well be acceptable, even slightly smaller values might be accepted, but the smaller the signal to noise ratio is the more experimental work the biologist will have to do in order to verify the findings of the motif discovery software.

Assume that we look for motifs by searching for strings in a given dataset, which satisfy a certain set of criteria. Then the *signal to noise ratio* is the ratio of the number of strings we find due to the fact that the motifs exist in the dataset over the number of strings that we can expect to find if the dataset is a random string. We typically do not know the number of strings that come from motifs. Instead we can determine the number of strings that satisfy given criteria, which is the number of strings due to the existence of motifs plus the random matches.

Let EPM be the expected number of matches we find due to the presence of the motif instances (given certain search criteria). For any search criteria let x be the number of different strings of length *l* that match the search criteria (amongst all possible strings of length *l*), the expected number of false positives is $EFP=x*n/4^l$ (the expected number of occurrences of any string of length *l* within a random string of length n is close to $n/4^l$). EFP is the number of strings we expect to find without any motifs being present and the input data being a random string.

We define the expected signal to noise ratio to be ESNR=EPM/EFP. EFP is obviously determined by the search criteria used in the algorithm. If we aim at finding all m motif instances, let *k* be the number of different motif instances, then EFP is at least (i.e. even for the best possible algorithm) $k*n/4^l$. Thus even the best algorithm is expected to return a set of at least $m+k*n/4^l$ motif instances. Using the PFM to characterise a motif, we need to be sure that the majority of the strings that define the PFM are actually motif instances. If the signal to noise ratio is low, we obviously include many strings, that are actually not motif instances, in the process of creating the PFM which means that the lower the SNR, the lower the quality of the PFM.

If we compare our approach (i.e. using Boolean matrices to model motifs) with the approach underlying the projection algorithm [1], we can see that we reduce the number of false positives significantly. This is expressed in Lemma 1 (below). Here we assume that the motif is perfectly describable by a Boolean matrix, i.e. each of its positions is either totally preserved or semi-preserved (with two bases appearing with equal probability).

Lemma 1:

Applying a BM with *sp* semi-preserved positions increases the expected signal to noise ratio by 2^{sp} compared to the projection algorithm.

Proof:

In the BM model, applying the correct BM to all input data, all m motif instances will be detected. Similarly selecting all fully preserved positions within the projection algorithm will find all m motifs. There are 2^{sp} different l-mers that match the BM, but there are 4^{sp} different l-mers that agree on all fully preserved positions in case we use the projection algorithm. Thus using the BM we expect to find $2^{sp}*n/4^l$ false positives, while we expect to find $4^{sp}*n/4^l$ false positives using the projection algorithm. Thus within BM we get $ESNR=m*4^l/(n*2^{sp})$ and for the projection algorithm we get $ESNR=m*4^l/(n*4^{sp})$. Thus for the projection algorithm the resulting expected signal to noise ratio is worse by a factor 2^{sp}.

Lemma 1 is the main reason for the fact that the algorithm BMA presented in this paper outperforms competitors that do not use BMs as their motif model.

2.3. Maximal impact enlargement

In the iterative algorithm presented below, we select in each of its iterations a set of strings K that we consider to be a set of motif instances. We iteratively enlarge this set by subsets of l-mers of the input data that have only one mismatch with the current Boolean matrix BM and in addition have high impact, defined as follows.

Definition:

Let $D:=D1_K$ (i.e. all appearances of all l-mers in the input data that have exactly one mismatch with BM_K). Let (i,b) be a position in BM_K such that $BM_K(i,b)=0$ and BM_K contains exactly one "1" in column i then $D_{i,b}$ is the subset of all appearances of strings from D that have a *b* in position *i* (here we restrict ourselves to motifs that have only preserved and semi-preserved positions). Amongst these sets we select the largest set (in case there are several sets of maximal size we select one of them randomly) and call it MIMP(K).

MIMP(K) has maximal expected signal to noise ratio amongst all $D_{i,b}$ as for all of these sets the expected number of false positives is identical.

3. The motif discovery algorithm BMA

The algorithm BMA starts with a single string that it regards to be a set of potential motif instances (a set of size 1). It expands this set of strings iteratively a fixed number of times (this number being the number of semi-preserved positions of the motif, often an estimated number) with sets of maximal impact. The sets of maximal impact are also the sets that have highest expected signal to noise ratio.

In realistic data (as demonstrated in Section 2.1.) these iterations will result only in a kernel (i.e. a proper subset) of a set of motif instances, but in the synthetic data used in simulations below (i.e. within the BM model) it will return all motif instances. In order to cope better with realistic data, we add a final stage to our algorithm, in which we select as additional candidates for motif instances further strings from all strings that are close (in terms of PFMs).

Before the algorithm BMA is applied, several parameters have to be selected: the motif length *l*, the number of semi-preserved positions *sp*, the number of kernels that will be selected in the first round of the algorithm *r*, the number of motif instances to be returned *mi* and the number of complete sets of complete motif sets to be *mc*.

The Boolean matrix algorithm for motif discovery (BMA):

1) For each *l*-mer in the input data firstly form a kernel K of size 1. Then repeat *sp* times: K:=K $\cup$ MIMP(K).
2) Amongst all sets K found in step 1, select the *r* largest.
3) For each of the *r* sets determine the PFM and amongst all *l*-mers from the input set select the *mi* strings with highest score.
4) Sort the *r* sets of motif candidates by the minimal score of their motif instances and select the *mc* sets with highest minimal score.

In Step 1) of BMA we from a kernel out of each *l*-mer of the input data (In [4] we have discussed alternatives to this approach). In each of the *sp* iterations of Step 1.) one preserved position is converted into a semi-preserved position. Step 2) selects the *r* "best" motif kernels (the size of the kernel is proportional to its expected signal to noise ratio). Step 3) expands the kernels, such that motif instances can be included that do not match the Boolean matrices in all positions. Step 4) selects the sets of motif instances that are least likely to have been generated randomly (the higher the score, the smaller the number of false positives that match the score).

4. Analysing the performance of BMA

4.1. Highest impact and highest SNR

BMA works by starting with a kernel of size 1, in fact it tries all strings from the input data of a fixed length, this includes all motif instances. Thus, if we happen to start with a kernel that consists of a motif instance, we start with a signal to noise ratio of infinity. Then BMA iteratively enlarges the kernel by including sets of strings with high expected signal to noise ratio.

Lemma 2:
Joining two disjoint sets S_1 and S_2 with expected signal to noise ratios $ESNR_1 = EPM_1/EFP_1$ and $ESNR_2 = EPM_2/EFP_2$ results in a set with expected signal to noise ratio

$EPRM_{1,2}=(EPM_1+ EPM_2)/(EFP_1+ EFP_2)$.

Proof: Trivial.

It is clear from Lemma 2, that as soon as we allow a low expected signal to noise ratio, we are including strings in the kernel, that are actually not motif instances. The more such strings are included in the kernel, the more likely it becomes that also in the next iteration many false positives are included and the algorithm diverges. In [4] we have demonstrated the impact using MIMP instead of including all strings of a given Hamming distance has on the expected signal to noise ratio. Here we only give the corresponding result.

Lemma 3:
If BMA is applied to a set of motif instances that belongs to the BM data model, then in every iteration of the BMA algorithm the ESNR is $(m/2^{sp})/(n/4^l)$.

Proof:
After iteration *s (s=1,...,sp)* of BMA there are *l-s* preserved positions in the corresponding BM. Thus for each of the not yet converted *sp-s* semi-preserved positions there are $2^s*m/2^{sp}$ motif instances that differ from the current kernel strings only in this position; while the number of false positives that satisfy the same criteria is $2^s*n/4^l$. This results in the ESNR given in the lemma.

Lemma 4:
Selecting from the set D1 (all strings that differ in exactly one of the (still) preserved positions) the subset MIMP improves the corresponding expected signal to noise ratio by a factor of at least $3l/sp$.

Proof:
The expected number of different strings that differ in exactly one position from the current matrix is $(3(l-s)*2^s+2*s*2^{s-1})*n/4^l$, while the number of corresponding motif instances is

*(sp-s)*2^s*m/2^sp*. The ESNR for BMA is given in Lemma 3. Thus the quotient of these two ESNRs is *(3l-2s)/(sp-s)*. It can easily be seen that this quotient is larger than 3l/*sp* for s≠0.

This proves lemma 4.

While Lemma 3 gives the expected signal to noise ration of every set that is joined to the kernel in each iteration of BMA, Lemma 4 states how much better this approach is compared to including all strings of Hamming distance 1 in any step.

4.2. Evaluating the solutions

After the algorithm has been applied to a large set of kernels we need to evaluate and rank the solutions. Different ways of evaluating solutions have been discussed in [4]. Within BMA the parameters *r*, *mi* and *mc* specify the ranking of the solutions. Each of these parameters is data dependent and will need finetuning in each application. Higher signal to noise ratio allows to make each of these three parameters smaller. The parameter *mi* is related to (and could be specified as such) the expected number of motif instances plus the expected number of false positives. Thus in many applications it will be possible to estimate *mi* reliably. We have chosen in the case of the SigmaB data *r*=20 and *mc*=10, but we have not done any analysis to justify this choice. More applications to real data are needed in order to see how critical the choice of these parameters is.

5. Experiment: Finding motifs in synthetic datasets

In order to substantiate our claims related to the performance of BMA we present some simulation results. As motif model we use BM, embedding 32 motifs in random strings of length 4^8=65,536. We vary sp, the number of semi-preserved positions, from 0 to 5. On the same data we apply also the projection algorithm. The projection algorithm will always find all 32 motif instances, as we try out all possible projections (typically this is not done as it is rather compute intensive). But in addition, as predicted via Lemma 1, it finds many more false positives than BMA and with low signal to noise ratio it often returns results that have nothing to do with the implanted motifs.

Table 1 below contain in the first column sp, the number of semi-preserved positions of the set of implanted motif instances. The second column (ENRPFM) is the expected number of returns if the correct PFM is found, which is $m+2^{sp}*n/4^l$. The third column (BMA result) gives the number of returns averaged over 10 experiments with different input data (both background and motif instances are generated randomly). The fourth column (ENRPRO) gives the expected number of returns for the projection algorithm provided that the projection coincides with the preserved positions of the PFM, these are $m+4^{sp}*n/4^l$. The fifth column (Projection results) is the average result over 10 experiments. The sixth column gives the ranking of the optimum result (ranked by the number of strings returned).

Table 1 (n=4^8):

sp	ENRPFM	BMA results	ENRPRO	Proj. results	Proj. rank
0	33	33	33	33	1
1	34	33	36	36	1
2	36	36	48	50	1
3	40	41	96	101	4
4	48	49 *	288	>300	> 10
5	64	68 **	1060	>1000	> 10

* : in one out of 10 runs no motif instances were amongst the first 20 returned strings (49is the average over the remaining 9 cases)

**: in two out of 10 cases no motif instances were found and in further two case only about half the number of instances were found (68 is the average over the remaining 6 cases)

If the number of returned strings is much higher than 32 (32 is the number of implanted motif instances) then it is probably quite difficult for the biological researcher to spot these motif instances and reject all the false positives. We also present the position amongst the ranking of all returned solutions, i.e. in case of BMA the correct solution is also always ranked highest but in case of the projection algorithm it happens that other projections have returned more candidates (and thus be ranked above the correct solution). They will be candidate sets that are regarded as more interesting initially, until it has been detected that they mainly contain false positives. The ranking for sp=4 and sp=5 is given as "below 10". This says that the 10 highest ranked solutions did not contain any motif instance, while the number of returned strings for these solutions were more than 300 for sp=4 and more than 1000 for sp=5. In both these cases BMA still returned meaningful results. We expect that further fine-tuning of BMA will lead to even better results.

As already mentioned in the introduction, we also did apply this new method to real datasets. We have been able to discover already reported motif instances in the genome sequences of bazillus subtilis and staphylococcus

aureus and several new instances most likely belonging to the transcription factor sigma B. A List of these new regulatory sequences can be found on [5].

6. Summary and further research

We have demonstrated in this paper that the algorithm BMA significantly outperforms in particular the well known projection algorithm [2]. One of the reasons for this is that it is based on and tailored towards the BM data model, which is more realistic than data models based on consensus strings and Hamming distance. Our main goal has been to show that we need to address the concept of signal to noise ratio in order to develop and finetune motif discovery algorithms. The algorithm BMA does just that:

- By choosing the motif kernels ,i.e. in the cases where we start with a motif instance we start with infinite signal to noise ratio.
- By the way motif kernels are enlarged, i.e. we make a small enlargement with a set that has highest signal to noise ratio.
- By the final stage of BMA, which selects the sets of motif candidates by the minimal score of all its members.

At the end it will be those researchers who are able to do corresponding biological experiments, who verify whether the sets of motif candidates that have been returned by a motif discovery algorithm are likely to be of biological value.

Close collaboration with the "Institut für Infektionsmedizin" at the Christain Albrechts Universität in Kiel we were able to apply the BMA algorithm to different sets of known motif instances and their sources. Here we detected a wide range of strings that are likely to be motif instances and had not been detected before.

It is (even though not dealt with in this paper) worth looking at how to reduce computation time. We could do so by reducing initially the dataset drastically (by using auxiliary information), i.e. we might know that even though the range of distances of the motifs from the start position of its gene is wide, most of the motifs are likely to be located in a much shorter interval in front of the gene. In this case we might firstly search for motifs only in this restricted area. This will also improve the expected signal to noise ratio as this area has a higher density of motif instances.

There is a wide range of further research needed in order to establish the results of this paper. BMA has to be applied to a wider range of synthetic data and more importantly to more real biological datasets. We also plan to fine-tune BMA in order to be able to handle datasets from a wider range of prokaryotes and eukaryotes. Partly this can happen via choosing the right parameters for BMA, but we also expect changes to the algorithm itself, in particular we expect to drop the restriction to Boolean matrices with at most two "1"s in each column. Through such generalization we expect to be able to get good performance for a wider range of motifs.

7. References

1. Buhler J, Tompa M: Finding motifs using random projections. *J Comput Biol* 2002, 9:225-242.
2. Thompson JD, Higgins DG, Gibson TJ: CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. *Nucleic Acids Res* 1994, 22:4673.
3. Petersohn A, Brigulla M, Haas A, Hoheisel J, Völker U, Hecker M: Global Analysis of the General Stress Response of Bacillus subtilis. *JOURNAL OF BACTERIOLOGY*, Oct. 2001, p. 5617–5631
4. Schröder J, Schröder H, Schimmler M, Tischer K: IGOM – Iterative generation of Position Frequency matrices, submitted to BIRD 2008. Available at this homepage: http://www.informatik.uni-kiel.de/fileadmin/arbeitsgruppen/technical_cs/Files-Jan/IGOM_paper.pdf
5. Search results of the BMA-algorithm on real datasets: http://www.informatik.uni-kiel.de/en/schimmler/staff/jan-schroeder/