IGOM – Iterative Generation of Position Frequency Matrices

Maximising Signal to Noise Ratio in Finding Motifs in DNA Sequences

Jan Schröder, Manfred Schimmler, Karsten Tischer

Christain Albrechts Universität, Kiel, Germany

Heiko Schröder

RMIT University / NICTA, Melbourne, Australia

Abstract:

Finding motifs in DNA sequences is one of the most demanding problems in computational biology. There are many algorithms available for motif discovery and none of them is totally satisfying. There is scope for improvement and we outline one way that leads to some improvements in discovering motifs particularly in cases where the motifs can be described adequately using position frequency matrices.

We introduce a new data model to specify kernels of motifs (i.e. significant subsets of the motif instances) in which all positions of a motif kernel are either 100% preserved (only one element is allowed in that position) or semi-preserved (only two different bases are allowed in that position). We call this model the SP-model.

Based on the SP model we have developed a motif discovery algorithm IGOM that attempts to keep the signal to noise ratio as high as possible, iteratively growing motif kernels, starting with kernels of size 1. In its final stage IGOM uses position frequency matrices as a scoring tool expanding the kernels by strings with lowest penalty.

We were able to verify the strength of IGOM comparing it to the projection algorithm with a range of synthetic data fitting the SP-model, and we also were able to demonstrate its ability in real datasets. Here we used it for real motif discovery in the cases of SigmaB in Bazillus Subtilise and Staphylococus Aureus.

1. Introduction

1.1 Background

In 2005 and 2006 Tompa et al [14,19] published analysis of motif finding algorithms and more recently Das and Dai [3] provided another survey, which we frequently refer to throughout this paper. Readers interested in getting an overview of the various approaches to motif finding are referred to these papers. One thing these surveys show clearly is that we urgently need better approaches, as the biological performance of all the algorithms that have been tested do not often meet practical requirements.

This paper attempts to add a new approach which combines ideas from the “word-based approaches” [2, 3, 11, 12, 13, 20] and ideas from the “probabilistic sequence models” [3, 1, 15], basing the motif search on the model of position weight matrices. We call our algorithm IGOM: **I**terative **G**eneration **o**f position frequency **m**atrices. In this paper, we focus on the problem of motif finding without using auxiliary information, as is done in many papers [2, 4, 5, 12, 16]. In addition we regard the assumption that we should find one motif each in a range of input sequences as auxiliary information (and do not make use of it in IGOM), while several approaches like Hidden Markov Models, MEME or Gibbs sampling rely substantially on this assumption [3]. Instead we only assume to know (or guess) the size of a motif and start by viewing every string of the input data as a candidate of being a motif instance and iteratively collect similar strings from the input data, building up a Position Frequency Matrix. At the end we only return those sets of detected motif candidates that satisfy certain criteria. These criteria are related to the likelihood that this set of strings would appear in a random input sequence. We show that our approach maximises the signal to noise ratio and thus reduces the pollution of the results that stems from false positives.

The projection algorithm developed by Buhler and Tompa et al [2] is regarded as one of the best algorithms for motif finding, without using auxiliary information. But we will show in Chapter 2 why it in many cases must fail when applied to real DNA sequences.

In [17] Hon and Jain present MaMF (a motif finding algorithms particularly tailored towards finding motifs in mammals and particularly successful). They similarly to our approach iteratively build up sets of motif instances starting from small motif kernels, but use a penalty function, that assigns the same penalty for every mismatch. Thus they are also not able to make use of the additional information contained in PFMs, e.g. in case of semi-preserved positions (i.e. two bases are allowed in this position) they give high penalties for one basis and no penalty for the other.

In this paper we give a close comparison to the projection algorithm and demonstrate on synthetic data that IGOM significantly outperforms the projection algorithm. As stated in [14] it is likely to be true that all current motif finding algorithms have strength and weaknesses and it is advisable to apply more than one of them. A study by Zaslavsky and Singh [17] shows that there are many real datasets where the projection algorithm is significantly more successful than Gibbs sampling and the MEME algorithm. Thus we compared IGOM only to the projection algorithm.

Our implementation of the presented algorithm doesn’t work on standard formatted input data like FASTA or GenBank yet so data has to be preformatted before the program run over it. But we are happy to give you the program anyway or run it ourselves over the data you might provide us with. Just send an email to jasc@informatik.uni-kiel.de.

1.2 Major features of IGOM

We base the theoretical analysis of the algorithms on the data model SP for motifs in which all positions within a motif are either preserved or semi-preserved – that means all instances of a motif have the same nucleotide in a certain position (preserved) or there are two different nucleotides to choose from (semi-preserved) (for further explanation see Section 2.1.). For the background data we assume random distribution throughout this paper. It could be considered incorporating other background models in order to increase the performance of IGOM further. The main characteristics of the IGOM algorithm are:

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

Applying IGOM to a dataset from Staphylococus Aureus [18] has shown that we were able to detect a range of previously unnoticed motif instances – a case where popular motif finding methods had failed.

1.3 Content

In Chapter 2 we present basic notation and techniques used in IGOM. In Chapter 3 we present IGOM and the way we choose seeds (kernels) to start the algorithm. Chapter 4 demonstrates that IGOM maximises the signal to noise ratio. Chapter 5 presents some simulation results that demonstrate that IGOM 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 motifs from a small set of strings K which we call a *motif kernel.*

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) [15] which 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 IGOM we model motifs using PFMs.

We use the PFM directly to score strings (in order to judge, whether they should be considered as motif instances or not). Let $x_1, x_2, \ldots, x_l$, be a string. Let PFM be a l by 4 matrix PFM(i,j) with $1 \leq i \leq l$ and j in {A, C, G, T}(see table 1 for an example of a PFM). 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. Maxscore for the PFM in Table 2 is 1000 (for Table 1 it is 946.2).

Several researchers have converted Position Frequency Matrices into Position Weight Matrices (PWM) for scoring purposes [15]. 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 SP

As examples for a PFM we show in Table 1 the matrix resulting from the data given in [18]. In table 2 we show a simplified version of this matrix, that we will use in order to explain some aspects of the algorithms. Of the 78 motif instances given in [18] 19 achieve the maximal score of this matrix, 31 have only one mismatch with the PFM of table 2 and only 5 motif instances have a mismatch in more than 2 positions. Thus table 2 is reasonably close to a description of a real set of motif instances.

This matrix belongs to a set of artificial data that we call the *SP model*. In it each motif position is either perfectly preserved (one basis has the value 100), or it is semi-preserved, i.e. two basis have the value 50.

In SP we use p for the number of preserved positions and sp for the number of semi-preserved positions, $p+sp=l$. Within SP the performance of motif finding algorithms can in some cases be analysed theoretically and it is particularly easy to calculate signal to noise ratios within this model.

Table 1: The PFM for SigmaB in bazillus subtilis (entries are percentages, the data is based on 78 motif instances

	1	2	3	4	5	6	7	8	9	10	11	12
A	0.0	6.4	3.8	2.6	56.4	51.3	10.3	0.0	3.8	43.6	100	39.7
C	0.0	1.3	0.0	2.6	9.0	10.3	0.0	0.0	3.8	2.6	0.0	9.0
G	100	1.3	5.1	1.3	12.8	10.,3	79.5	98.7	91.,0	1.3	0.0	9.0
T	0.0	91.0	91.0	93.6	21.8	28.2	10.3	1.3	1.3	52.6	0.0	41.0

Table 2: Synthetic PFM with similarity to SigmaB

	1	2	3	4	5	6	7	8	9	10	11	12
A	0	0	0	0	50	50	0	0	0	50	100	50
C	0	0	0	0	0	0	0	0	0	0	0	0
G	100	0	0	0	0	0	100	100	100	0	0	0
T	0	100	100	100	50	50	0	0	0	50	0	50

In columns 5, 6 and 7 the synthetic matrix of Table 2 diverges most from the real data of Table 1. There are entries of over 10 that have been converted into 0s.

Let Z_i be the number of zero entries in column i of a PFM of the type shown in Table 2 and let N_i be the number of non-zero entries in column i, $(N_i+Z_i = 4)$. The number of different strings that score maximally for this matrix is $\prod_j N_j$

Lemma 1:

In the SP model the number of different strings that score in all but one column is

$\Sigma_i (Z_i * \prod_{j\neq i} N_j)$.

Proof:
This follows directly from the fact that there are Z_i positions to choose from which don't match the matrix in column i and the product term is the number of the strings that match the rest to combine with.

Thus it is very simple to calculate the expected number of false positives within the SP model (see Section 2.2).

2.2. Signal to noise ratio in motif finding 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 finding 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 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 finding software.

Assume that we look for motifs by searching for strings in a given dataset, which satisfy a certain set of criteria. Then the expected *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 $k \leq m$ 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 in the process of generating the PFM 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, which are actually not motif instances, in the process of creating the PFM. If in the iterative process the PFM gets “distorted” too much, i.e. giving high scores to strings that are not motif instances, the PFM does not converge anymore to a correct specification of a motif.

Lemma 2:

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

Proof:

In the SP model, applying the correct PFM to all input data, all m motif instances will be detected. As there are 2^{sp} different l-mers that match the PFM we would in addition expect to find $2^{sp} * n/4^l$ false positives, resulting in $ESNR = m * 4^l / (n * 2^{sp})$. Using a projection algorithm on $p = l - sp$ positions and having found the correct *p* positions, will also find all m motif instances in the input string, but we would expect to find $4^{sp} * n/4^l$ false positives. Thus for the projection algorithm the resulting expected signal to noise ratio is by a factor 2^{sp} worse.

Lemma 2 is the main reason for the fact that the IGOM algorithm presented in this paper outperforms competitors that do not use PFMs as their motif model. Lemma 2 shows how the advantage of IGOM over the projection algorithm depends on the number of semi-preserved positions in the motif.

2.3. Maximal impact change

The algorithm presented below develops a PFM of the hidden motif iteratively. This is done by admitting new strings to the set of candidates each round. To keep the SNR at the highest possible value throughout the algorithm only very few *different* strings are admitted to form the PFM because every string brings its own probability of false positives. So the strings taken into the matrix have to be chosen carefully.

So each round we chose a set candidates which has the highest expected SNR by analysing the *impact* of positions in the matrix:

Let K be the set of strings from the dataset that score in every column of the PFM but one (strings that score in every column are already part of the matrix).

Let $K_{i,b}$ be the set of all strings from K that have a b in position i. Let $max_i(PFM)$ be the maximal value in row i of PFM. We define the impact $IM(i,b,K)$, $b \in \{A, C, G, T\}$ as $IM(i,b,K)=|K_{i,b}|*|max_i(PFM) - PFM(i,b)|$. We call the set $K_{i,b}$ with highest impact $MIMP(K)$.

So the impact of a set of strings $K_{i,b}$ is mainly defined by its size. The above definition of impact gives high priority to changing a preserved position into a semi-preserved position, as in this case the difference $|max_i(PFM) - PFM(i,b)|$ is highest. The more strings there are that mach the current PFM in all but one position that have the same nucleotide (position and base) in common the better will be the SNR for this set. Note that $K_{i,b}$ is not a set that includes the same string over and over again. The strings can differ as long there are semi-preserved positions in the PFM.

This leaves to consider if all possible strings in $K_{i,b}$ should be admitted to the matrix or only those which have a high number of occurrences in the dataset to maximize the SNR. But our experience with real motif data shows that every possible instance of the PFM belonging to a motif will be taken.

The algorithm IGOM, presented next, expands the motif kernel K using the set $MIMP(K)$. This way we expect to find an additional semi-preserved position in each iteration.

3. The motif finding algorithm IGOM

The algorithmic scheme of IGOM is that it starts with a set K of strings that it regards to be a set of potential motif instances. It expands this set of strings iteratively with further strings that have high similarity to the current set. The similarity is expressed via a threshold *th* on the score these strings have to achieve.

This algorithmic scheme can be implemented in a range of ways. We have chosen an implementation of this scheme that attempts to maximise the signal to noise ratio, i.e. reduces the number of false positives as much as possible, given the SP model.

We now present the algorithm that we use for motif finding, which can be tuned further in particular in cases where auxiliary information is available.

We use the following algorithm, IGOM:

```
1.) Select a motif kernel K. $s:=1$. Select a function size(s) that limits the number of motif
    candidates in each iteration of the algorithm s. Select th as a threshold used to
    terminate the iterations.
2.) While $SNR(K)>th$
      a. Determine the $PFM(K)$.
      b. $K_{old} := K$.
      c. $K$:= set of the size(s) strings which score in every except of one position in the
         PFM.
    d. $K:= K_{old} \cup MIMP(K)$.
    e. s:=s+1
3.) Amongst all strings that score in each of the $l$ positions determine the $m+EFP$ strings
    with highest score.
```

The first step initializes the Algorithm (more details on this step follow in the next section), while the third gathers the results and finishes the Algorithm. The main work is done in the second step. The position frequency matrix is developed iteratively by including a set of candidates that is estimated to have the best signal to noise ratio as described above (d.). The fact that K is chosen amongst strings that don't match the PFM in every position (c.) ensures that K and K_{old} are always disjoint and no candidates that have already been taken to form the PFM will contribute a second time.

3.1. Finding motif kernels

We call a set of strings that we suspect to belong to the same motif a *motif kernel*.
The selection of motif kernels in the first step of the algorithm can be executed in different ways and is dependent on the parameters of the dataset as well as the expected parameters of the motif. A motif kernel might be given to us, created by biological experiments.
If $m/2^{l-p}>1$ then we could use as motif kernels all strings of length l that have at least t occurrences in the data set (t = m/2^{l-p} is the threshold, it is important to note that we do not know m and p in advance, but we might sometimes have good guesses based on auxiliary information. Thus we should use this approach whenever we find l-mers that appear significantly more often than $n/4^l$ times).

We can instead use every l-mer of the input dataset as 1-element kernel. Depending on the size of the input data this approach might not always be practical, but this is what we have done in most of our experiments.

4. Analysing the performance of IGOM

4.1. Highest impact and highest SNR

IGOM works by starting with a small kernel, in fact it tries all strings of a fixed length from the input data, thus if this length is smaller or equal to the length of the motifs we try to find, this will include all motif instances. So 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 IGOM iteratively enlarges the kernel by including sets of strings with high expected signal to noise ratio.

Lemma 3:
Joining two disjoint sets S_1 and S_2 with expected signal to noise ratios EPM_1/EFP_1 and EPM_2/EFP_2 results in a set with expected signal to noise ratio
$(EPM_1+ EPM_2)/(EFP_1+ EFP_2)$.
Proof:
The proof follows obviously from the fact that the two sets are disjoint so both the positives and the negative (false positive) matches will add up.

It is easy to see that as long as we limit the expected signal to noise ratio of all joining sets of strings using a lower limit for the expected signal to noise ratio, we will never fall below this lower limit for the kernel. It is also clear from Lemma 3, 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. These will then (step 2. (a) in IGOM) contribute to the entries in the intermediate PFM and thus make it more likely that we will continue to include “wrong” strings, “damaging” the PFM even further. In the remainder of Section 4.1. we use these simple facts in order to motivate the use of MIMP within IGOM.

We could take the approach to search for all strings that have Hamming distance 1 from any existing string, as it can be expected that within a set of motifs there is one motif instance that has several motif instances within Hamming distance 1. But also the number of random matches with these strings increases. Thus this approach is only bound to be successful if the signal to noise ratio is sufficiently high.

In the following we assume that the set of motif instances fits the SP model with sp semi-preserved positions. At the start of the algorithm we select a single string as motif kernel. If this string is a motif instance (as IGOM selects all strings of the input data, this will sometimes be the case), then there are $sp*m/2^{sp}$ motif instances with exact Hamming distance 1 from the given string and the expected number of strings in the input sequence that have exactly Hamming distance 1 from the given string is $3l*n/4^{l}$. This results in an ESNR of $(sp*m/2^{sp})/(3l*n/4^{l})$. Any particular motif instance at hamming distance 1 exists $m/2^{sp}$ times and we have to expect $n/4^{l}$ matching false positives that match this motif instance. Thus by restricting the expansion of the kernel by only including one additional motif instance we get an ESNR of $(m/2^{sp})/(n/4^{l})$ which is an improvement of the expected signal to noise ratio by a factor of $3l/sp$. This restriction is performed in step 2.e. of the IGOM algorithm. It can be shown that this improvement increases in every iteration of IGOM. The following Lemma states that in each iteration of IGOM the ESNR is maintained at the same level.

Lemma 4:
If IGOM is applied to a set of motif instances that belongs to the SP data model, then in every iteration of the IGOM algorithm the ESNR is $(m/2^{sp})/(n/4^{l})$.
Proof:
After iteration s of IGOM there are $l-s$ preserved positions in the corresponding PFM. 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 5:
Moving from the set of all strings that differ in exactly one of the (still) preserved positions to a set that corresponds to converting exactly one preserved position to a semi-preserved position 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 IGOM is given in Lemma 4. 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 *$s \neq 0$*. This proves lemma 5.

While Lemma 4 gives the expected signal to noise ration of every set that is joined to the kernel in each iteration of IGOM, Lemma 5 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.

Methods of ranking the solutions have been discussed in the literature. A most recent paper discussing these issues is [6] which uses multiple alignment techniques.

We use two different approaches. The first method simply uses the score of their *m-th* candidates, where *m* is the expected number of motif representatives. Let the solution be a triple *(PFM, C, th)* where *PFM* is a position frequency matrix and *C* is a set of strings from the input data with each string scoring at least *th* with the given PFM. Let *sm* be the score of the *m-th* element of *C* (*C*, sorted by the score in descending order). For a given PFM increasing *th* coincides with reducing the probability of matching the score with a random sequence. But it is not necessarily the case that two solutions with the same score of the *m*-th candidate are equivalent, as can be demonstrated by simple examples (e.g. it is possible to construct two PFMs with identical maximal score but significantly different numbers of possible matches). The major advantage of this method is that it requires almost no additional computation.

A more precise and reliable measure is to evaluate each triple *(PFM, C, th)* by the ratio *$|C|/RSS$*, where *RSS* is the expected number of sequences matching the score for a given n (assuming that the input is a random sequence). *$RSS=TSS*n/4^l$*, with *TSS* being the total number of different sequences matching *th*. Since *$n/4^l$* is constant for a given motif finding problem, we can use *$|C|/TSS$* as evaluation criteria. In order to compute *TSS* we need to score all possible sequences of length *l,* these are *4^l* sequences. This can be done in acceptable time for significant problem sizes.

5. Experiment: Finding motifs in synthetic datasets

In order to substantiate our claims related to the performance of IGOM we present some simulation results. As motif model we use SP, 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 it finds many more false positives than IGOM and with low signal to noise ratio it often returns results that have nothing to do with the implanted motifs.

Tables 3 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 (IGOM 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 3 ($n=4^8$):

sp	*ENRPFM*	*IGOM results*	*ENRPRO*	Projection results	Projection rank
0	33	33	33	33	1
1	34	34	36	36	1
2	36	36	48	50	1
3	40	41	96	101	4
4	48	49 (2x not all found)	288	>300	below 10
5	64	23 with 20 motif instances	1060	>1000	below 10

If the number of returned strings is much higher than 32 (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 IGOM 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 IGOM still returned meaningful results. We expect that further finetuning of IGOM will lead to even better results.

6. Summary and further research

In this paper we have demonstrated that the algorithm IGOM 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 SP 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 finding algorithms. The algorithm IGOM does just that: By choosing the motif kernels, by the way motif kernels are enlarged and by the way termination criteria are defined, that make sure that sets of strings that are most unlikely to be found in random data are kept as candidate sets. At the end it has to be the microbiologist who is able to do corresponding biological experiments, who verifies whether the sets of motif candidates that have been returned by a motif finding algorithm are likely to be of biological value.

Close collaboration with the "Institut für Infektionsmedizin" at the Christian-Albrechts-Universität in Kiel we were able to apply the IGOM 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 findings of this paper. IGOM has to be applied to a wider range of synthetic data and more importantly to real datasets. We also plan to finetune IGOM in order to be able to handle datasets from prokaryotes, eukaryotes and viruses. The SP model introduced in the paper can be further generalised and the motif finding strategy can be adopted to this generalisation. Through this generalisation we expect to be able to get good performance for a wider range of motifs.

7. References

1. Bucher P: Weight matrix description for four eukaryotic RNA polymerase II promoter element derived from 502 unrelated promoter sequences. *J Mol Biol* 1990, 212:563-578.
2. Buhler J, Tompa M: Finding motifs using random projections. *J Comput Biol* 2002, 9:225-242.
3. Das M, Dai H, A survey of DNA motif finding algorithms, *BMC Bioinformatics* 2007, 8(Suppl 7).
4. Eskin E, Pevzner P: Finding composite regulatory patterns in DNA sequences. *Bioinformatics* 2002, 18(Suppl 1):S354-S363.
5. Hon LS, Jain AN: A deterministic motif finding algorithm with application to the human genome. *Bioinformatics* 2006, 22:1047-1054.
6. Kankainen M, Löytynoja A, MATLIGN: a motif clustering, comparison and matching tool, *BMC Bioinformatics* 2007
7. Lawrence CE, Reilly AA: An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins* 1990, 7:41-51.
8. Lawrence CE, Altschul SF, Boguski MS, Liu JS, Neuwald AF, Wootton JC: Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science* 1993, 262:208-214.
9. Pevzner P, Sze S: Combinatorial approaches to finding subtle signals in DNA sequences. *Proceedings of the Eighth International Conference on Intelligent Systems on Molecular Biology, San Diego, CA* 2000, 269-278.
10. Sinha S, Tompa M: A statistical method for finding transcription factor binding site.

Proceedings of the Eighth International Conference on Intelligent Systems on Molecular Biology, San Diego, CA 2000, 344-354.

11. Sinha,S. and Tompa,M. (2003) YMF: A program for discovery of novel transcription factor binding sites by statistical overrepresentation. Nucleic Acids Res., 31, 3586-3588.

12. Tompa M: An exact method for finding short motifs in sequences, with application to the ribosome binding site problem. *Proceedings of the Seventh International Conference on Intelligent Systems on Molecular Biology* 1999, 262-271.

13. Tompa M: Identifying functional elements by comparative DNA sequence analysis. G*enome Res* 2001, 11:1143-1144.

14. Tompa M, Li N, Bailey T, Church GM, De Moor B, Eskin E, Favorov A, Frith MC, Fu Y, Kent WJ, Makeev VJ, Mironov AA, Noble WS, Pavesi G, Pesole G, Regnier M, Simonis N, Sinha S, Thijs G, van Helden J, Vandenbogaert M, Weng Z, Workman C, Ye C, Zhu Z: Assessing computational tools for the discovery of transcription factor binding sites. *Nat Biotechnol* 2005, 23:137-144.

15. 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.

16. van Helden J, Andre B, Collado-Vides J: Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J Mol Biol* 1998, 281:827-842.

17. Zaslavsky E, Singh M: A combinatorial optimization approach for diverse motif finding applications, *Algorithms for Molecular Biology* 2006, 1:13, 2006

18. 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

19. Li N, Tompa M: Analysis of computational approaches for motif discovery, *Algorithms for Molecular Biology* 2006, **1:**8

20. Pevzner A, Sze S-H: Combinatorial Approaches to Finding Subtle Signals in DNA Sequences. *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, 269-278, 2000.