UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Evidenčné číslo: 77976530-abea-4880-9ee6-849dc17b092a

Discovering motifs in mitochondrial DNA

2011 Bc. Jaroslav Budiš

UNIVERZITA KOMENSKÉHO V BRATISLAVE

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Discovering motifs in mitochondrial DNA

Diplomová práca

Študijný program: Aplikovaná informatika

Študijný odbor: 9.2.9. aplikovaná informatika

Školiace pracovisko: Katedra aplikovanej informatiky

Vedúci diplomovej práce: Mgr. Broňa Brejová, PhD.

Bratislava 2011 **Bc. Jaroslav Budiš**

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta:	Bc. Jaroslav Budiš
Študijný program:	aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor:	9.2.9. aplikovaná informatika
Typ záverečnej práce:	diplomová
Jazyk záverečnej práce:	anglický

Názov: Discovering Motifs in Mitochondrial DNA

Cieľ: The goal of the thesis is to implement a new software tool for motif discovery in DNA sequences specifically adjusted for specific properties of yeast mitochondrial genomes.

Vedúci: Mgr. Bronislava Brejová, PhD.

Dátum zadania: 18.10.2010

Dátum schválenia: 09.11.2010

doc. Ing. Igor Farkaš, PhD.
garant študijného programu

študent

vedúci

Acknowledgement

I would like to thank Dr. Broňa Brejová for her valuable guidance and patience during the completion of this thesis

Jaroslav Budiš

Declaration on word of honour

I declare on my honour that this work is based only on my own knowledge, references and consultation with my supervisor(s).

Bratislava 6. 5. 2011 . . . .

Jaroslav Budiš

Abstrakt

BUDIŠ, Jaroslav: Hľadanie motívov v mitochondriálnej DNA. Diplomová práca. Univerzita Komenského v Bratislave; Fakulta matematiky, fyziky a informatiky; Katedra aplikovanej informatiky. Bratislava (2011), 57 s. Školitel: Mgr. Broňa Brejová, PhD.

Úsek, ktorý sa vyskytuje opakovane v biologických sekvenciách, nazývame motív. V práci podrobne analyzujeme algoritmus na hľadávanie motívov MEME, ktorý sme aj implementovali. Navrhujeme novú metódu inicializácie parametrov modelu pre EM algoritmus, ktorý je jadrom MEME algoritmu. Výsledkom je zníženie počtu iterácií v porovnaní s prístupom, ktorý používa MEME. Algoritmus bol následne upravený na úlohu vyhľadávania motívov v mitochondriálnych genómov vyznačujúcich sa špecifickou štruktúrou. Genómy s vysokým zastúpením adenínu a tymínu obsahujú oblasti, v ktorých prevláda cytozín a guanín. Tie sú tiež nazývané GC ostrovy. Nájdenie motívov v GC ostrovoch môže viesť k ich rozdeleniu do skupín podľa príbuznosti.

Kľúčové slová:

hľadávanie motívov, MEME algoritmus, štruktúra mitochondriálneho genómu, GC ostrov

Abstract

BUDIŠ, Jaroslav: Discovering motifs in mitochondrial DNA. Master Thesis. Faculty of mathematics, physics and informatics. Comenius University, Bratislava (2011), 57 p. Supervisor: Mgr. Broňa Brejová, PhD.

Motif discovery problem abstracts the task of discovering conserved cluster of similar, relatively short substrings in set of biological sequences. We analyse in depth motif discovery algorithm MEME which we have also reimplemented. We propose a new method for initializing the iterative EM algorithm which is a core algorithm of the MEME. In our tests it decreases the number of required iterations compared to a the approach that is used in MEME. We also propose additional changes to deal with specific structure of mitochondrial genomes. In particular, we adjusted MEME model to account for the existence of GC islands, which are relatively short areas significantly enriched in cytosine and guanine. Discovery of patterns in these regions may eventually lead to their classification and therefore provide evidence of their conservation.

Keywords:

motif discovery, MEME algorithm, structure of mitochondrial genome, GC island

Contents

List of Tables

List of Figures

Introduction

A motif is a short region in biological sequences that occurs more frequently than expected by chance. Enrichment of sequences with similar elements may indicate that they share some function or evolutionary origin. Discovery of motifs is thus an important task in current molecular biology. It has application in identification of gene expression regulatory network, multiple sequence alignment and protein structure and function prediction.

Our motivation is to find motifs in CG islands that are specific features in mitochondrial genomes of yeasts which differ structurally from the rest of the genome. This problem is analogous to the motif discovery problem thus we chose the motif discovery algorithm MEME and modified it to this task. In particular, we have adjusted the probabilistic model underlying MEME to take into account statistical differences between GC clusters and surrounding sequence. The motif discovery process is then directed to the GC cluster areas to find significant motifs within them. Finding motifs in the GC islands may eventually lead to their classification to several evolutionarily related families.

In the first chapter we introduce necessary biological background and an example of an application of the motif discovery, specifically the identification of regulatory sites for gene expression. We also review various models for motif representation as well as algorithms that have been proposed for motif discovery.

In order to introduce proposed MEME modifications, it is necessary to understand the main concepts that stands behind the algorithm. We introduce them in detail in the second chapter. We describe the core algorithm, the expectation-maximisation (EM) algorithm, as well which is an iterative process, that improves the model parameters in each iteration. The EM algorithm converges to a local maximum, choosing appropriate initial parameters is therefore a key condition to find the global maximum, and hence the best motif. The MEME algorithm initialises parameters from random samples from the input dataset. In third chapter we introduce a new way of selecting initial parameters that leads to decrease in the number of required iterations thus potentially to better speed of the whole algorithm.

In the fourth chapter we introduce modifications of the MEME algorithm that were made in order to discover similarities in the CG islands and study their impact on synthetic and real data.

Chapter 1

Motif discovery algorithms

1.1 Biological background

A gene is a fundamental unit of inherited information in living organisms. It can be described as a continuous stretch of nucleotides located in deoxyribonucleic acid (DNA) that serves as template for the copying process called transcription.

An important goal in current molecular biology is to understand the cellular systems that participate in transcription and a subsequent process called translation that produces proteins (1.1). One of the subgoals is to find units that are responsible for regulating gene expression under different environmental conditions.

1.1.1 Gene expression

The main idea is that gene expression starts by binding of specific proteins, known as transcription factors to promoter and enhancer regions, usually located before a region of DNA containing a gene. Bound proteins can regulate gene expression by promoting (activator) or blocking (repressor) recruitment of RNA polymerase, the essential process that starts transcription of genetic information from gene into messenger RNA (mRNA).

Messenger RNA carries coding information to the sites of protein synthesis and therefore serves as blueprint for protein product. After further processing, the information is translated to a chain of amino acids forming a protein. Proteins are main actors in biolog-

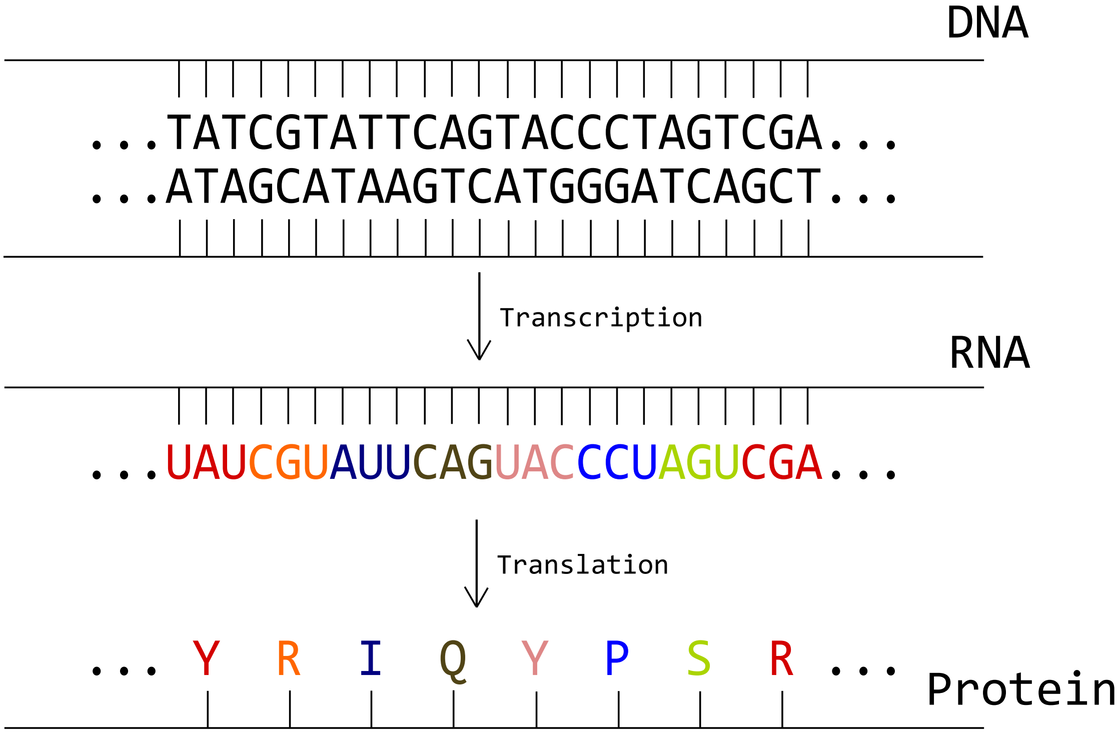
DNA
...TATCGTATTCAGTACCCTAGTCGA...
...ATAGCATAAGTCATGGGATCAGCT...
Transcription
RNA
...UAUCGUAUUCAGUACCCUAGUCGA...
Translation
... Y R I Q Y P S R ...
Protein

Figure 1.1: Gene expression

ical processes within cells and are required for their structure, function, and regulation.

1.1.2 Transcription factor binding sites

Identifying of regulatory sites, especially binding sites of transcription factors is a typical application of the pattern discovery. Due to difficulty in accurately assaying protein-DNA interaction on a large scale, various computational methods have been proposed for discovering DNA sequences required for proper binding of such proteins. Profiles discovered by these methods can serve as good candidates for further *in vitro* experiments to show evidence of their binding potential.

Discovery of binding sites typically starts with selecting putatively co-regulated genes. These co-regulated sets are often obtained by using clustering (1) to identify genes that share same functional category or experimentally, by identify genes that are expressed under a number of different environment conditions. It is assumed, that expression of the genes from the same category is regulated by a common regulatory network which should contain binding sites of same transcription factors. Motif discovery is therefore performed on the relevant promoter regions of co-regulated genes.

1.2 Motif representation

A DNA motif can be defined as a sequence pattern composed of several nucleotides that has some biological significance, for example transcription factor binding site. It is worth noting that these motifs are often relatively short, usually between 5 to 20 base pairs (bp). In addition, their occurrences do not have to be necessarily identical, what makes localising them even more challenging. Various models for motif representation have been proposed. They differ in various aspects. Most important is the balance between simplicity and representation power. Multiple algorithms have been developed to search for motif based on chosen representation.

1.2.1 Word-based representation

Perhaps the simplest form of motif representation is a *consensus sequence*. In this case, a motif is simply a word composed of letters that indicate preferred nucleotide at each position of the motif. Nucleotides are encoded by their initial letter. The codes are presented in the table 1.1.

Example of such a representation is word CACGTG for motif *Arnt* obtained from the Jaspar database (2). The matches of the motif in a sequence offer good candidates for binding sites, however we need to be aware of possibility that such sequences could also arise at random.

This approach benefits from its simplicity, however lacks some expressive power required to represent motif degeneracy. High-quality motifs obtained from *in vitro* experiments indicate that some positions in motif can be occupied by nucleotides of various types without significant impact on binding ability of such binding site. The strict *consensus sequence* model is not able to deal with this ambiguity, so extension of this representation is required. Various modifications of strict word representation have been proposed to allow certain degree of flexibility in the motif.

One of the solution of this problem is to allow a certain number of mismatches between motif and its occurrence in the sequence. Parameter k determines allowed Hamming distance between sequence match and pattern. In other words, sequence S matches pattern P with at most k mismatches if exists such substitution of at most k nucleotides in S that final sequence is equal to P.

This approach suffers for several reasons. The most apparent of them is that failure at each position of motif is considered equally important. However collected motifs indicate that some positions in motif are more conserved and therefore more significant for functionality of the binding site (3; 4). Similar problem is among possible changes at a particular motif position. Some nucleotide types may be mutually substitutable at this position without impact on functionality, however assignment of certain nucleotides can disable the binding site. There is no way to address this feature in such representation without modifications.

To deal with these problems, another extension has been proposed. The idea is to incorporate information about degeneracy into ambiguous codes. Each code is represented by single unique character and corresponds to a subset of nucleotides. Code then matches any character from associated subset. For example pattern A[CG]A matches only sequences ACA and AGA.

Codes for all possible subsets have been established by the International Union of Pure and Applied Chemistry (IUPAC). The codes are presented in the table 1.1.

Another relaxation from the strict consensus sequence representation is to allow gaps between certain motif positions. This can be done by using ambiguous code for any nucleotide, N. For example code CNNG matches region, where between cytosine and guanine are located exactly two arbitrary nucleotides.

Another notation is required to address flexible number of gaps. The pattern $x(i, j)$ matches any string of nucleotides of length between i and j, for example $A - x(2, 4) - C$ is same as combination of patterns $ANNC$, $ANNNC$ and $ANNNNC$.

Multiple modifications can be joined to provide a more powerful motif model. For example, the *PROSITE* database (5) uses complex model for protein sequences containing flexible gaps and ambiguous codes.

Code	Allowed nucleotides	Description
A	A	**A**denine
C	C	**C**ytosine
G	G	**G**uanine
T	T	**T**hymine
U	U	**U**racile
R	G A	pu**R**ine
Y	T C	p**Y**ramidine
K	G T	**K**etone
M	A C	a**M**ino group
S	G C	**S**trong interaction
W	A T	**W**eak interaction
B	G T C	not A (**B** comes after A)
D	G A T	not C (**D** comes after C)
H	A C T	not G (**H** comes after G)
W	G C A	not T, not U (**V** comes after U)
N	A G C T	a**N**y

Table 1.1: IUPAC code table

1.2.2 Position weight matrix

Another widely used type of motif is a position weight matrix (PWM), also known as a position-specific weight matrix (PSWM) introduced by Gunnar von Heijne (6). The main advantage of this representation is that it allows to express belief about significance of each nucleotide at each motif position.

Motif is represented as a $w \times L$ matrix f, where w is the width of the motif and L is the size of the alphabet, that is four distinct nucleotides for DNA sequences. Positive score f_{ip} means that nucleotide type indexed by p is preferable on i-th position of motif. On the other hand negative score penalises sequences that have nucleotide p at the i-th position. Overall score for a sequence is computed by summing scores for each position within the motif window. The higher the score, the better chance that the sequence is an occurrence of the motif.

Decision if a PWM matches a sequence is usually made by setting up some threshold. The motif matches each sequence with score higher than the threshold. It is necessary to choose an appropriate threshold to balance between the number of false positives and false negatives. If selected value is too small, many positions in DNA would be marked as occurrence of motif, even some false random subsequences. However, if high value is chosen, some weak sites may not reach the required score and thus will not be discovered.

A closely related representation models a motif as a matrix of nucleotide probabilities on each position. Probability that a sequence is an occurrence of a motif can be easily calculated as a product of probabilities of nucleotides present at each position of the motif. Comparison with the probability that the sequence comes from a genomic background provides a degree of belief about binding ability of the examined sequence.

Motifs represented as nucleotide probabilities can be easily visualised using sequence logos (7). A sequence logo consists of ordered stacks of letters where each stack shows a distribution of nucleotides at one motif position. The height of each letter of the stack is proportional to its frequency, and the letters are sorted so that the most common one is on top. The height of the entire stack is then adjusted to signify the information content of the sequences at that position. The example of a sequence logo obtained from the Jaspar database for the following frequency matrix of transcription factor ZEB1 is at figure 1.2.

$$\begin{array}{c c} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{array}{c} A \\ C \\ G \\ T \end{array} & \begin{pmatrix} 0.024 & 0.927 & 0.000 & 0.000 & 0.000 & 0.244 \\ 0.829 & 0.000 & 0.976 & 0.927 & 0.024 & 0.024 \\ 0.024 & 0.049 & 0.024 & 0.073 & 0.000 & 0.390 \\ 0.122 & 0.024 & 0.000 & 0.000 & 0.976 & 0.341 \end{pmatrix} \end{array}$$

A PWM is a powerful method for motif representation thanks to good balance between simplicity and expression power. One simplification is the assumption that each motif position is independent of the nucleotides observed at other positions. However dependencies between nucleotides at different motif positions have been observed (8; 9). Several algorithms have been proposed using a generalized PWM with incorporated information about

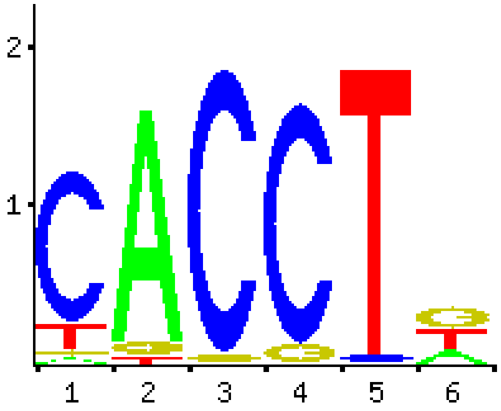

Figure 1.2: Sequence logo representation of protein ZEB1

pairwise dependencies (10).

Another problem is the absence of gaps in the model. An motif occurrence may lack some of the motif positions or contain several nucleotides between two consecutive positions and still function properly. These features can be expressed by a more complex scoring scheme, such as the one proposed in (11). However it is possible to directly extend PWM model with insertions and deletions. Such models are called profile hidden Markov models.

1.2.3 Hidden Markov model

A hidden Markov model (HMM) is a probabilistic model with finite set of states. In a particular state, an outcome can be generated, according to the associated probability distribution. Transitions among the states are governed by a set of probabilities called transition probabilities. Only outcome is visible to an external observer, the order of states that have generated sequence is hidden.

HMMs were introduced into computational biology by Gary A. Churchill in (12). Notice that it is possible to express position weight matrix representation as an HMM with by w match states $s_1, s_2 \ldots s_w$, where state s_i generates exactly one nucleotide from distribution f_i and moves to state s_{i+1}. Model starts in state s_1 and finishes after generating exactly w nucleotides. Model representing binding sites of ZEB1 protein (figure 1.2) is

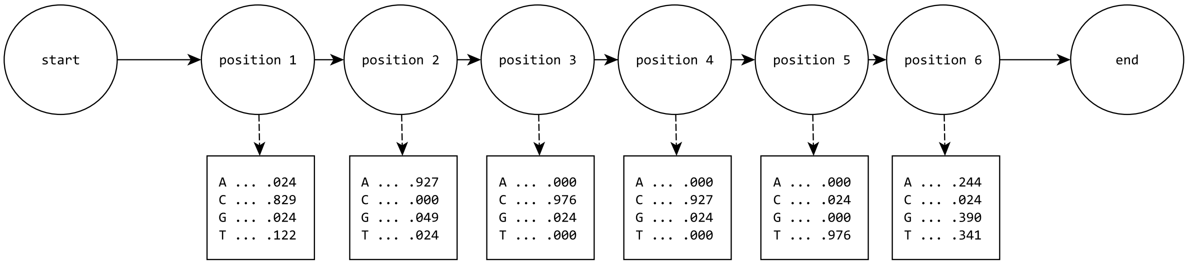

Figure 1.3: Hidden Markov model for binding site of the ZEB1 protein

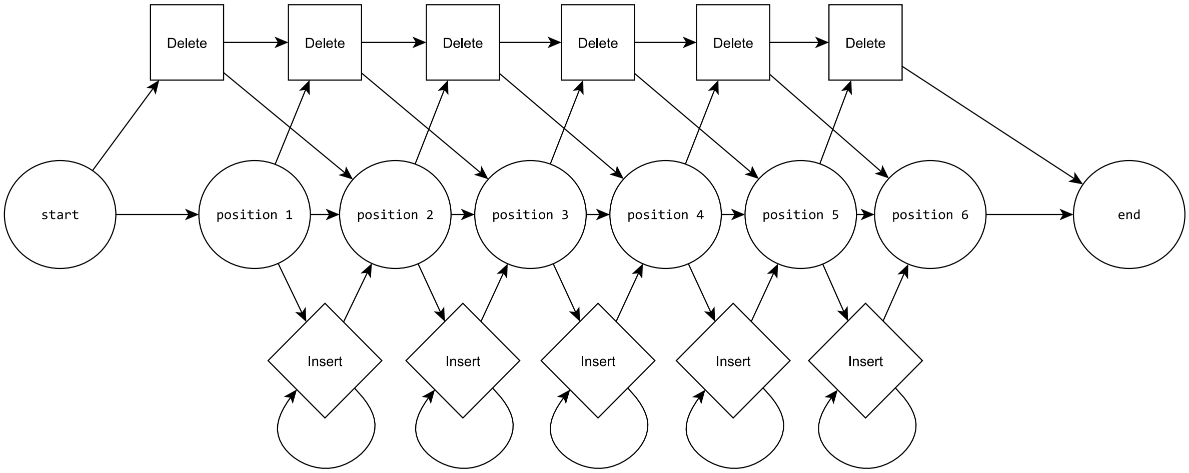

Figure 1.4: Profile hidden Markov model

presented in figure 1.3.

Main benefit earned by this representation is simple incorporation of gaps directly into model. It is achieved by addition of special insertion and deletion states. HMMs with these states are called profile hidden Markov models (pHMM) and can be observed in figure 1.4.

Profile HMMs have a high expression power, however they still assume position independence. Another problem is the high number of parameters that have to be estimated, and thus more data is required to establish a satisfying model. When only a limited number of samples is provided, which is quite common, a complex model may overfit the data and later provide worse results for new samples.

1.3 Enumerative methods

Enumerative methods for solving the motif discovery problems are typically based on word-based motif representation. The methods differ mainly in the expression power of used motif model. It varied from the plain consensus sequence to the more complex models which enable a certain level of a motif degeneracy as presence of gaps or ambiguous nucleotides.

1.3.1 Exhaustive search

Perhaps the simplest solution based on the word-based is exhaustive search. First, the appropriate motif model is selected with defined level of degeneracy. Then a set $\mathcal{C}$ is filled with all possible motifs that satisfy model constraints. In the case of the simple consensus sequence without degeneracy they could be all strings of user defined length w. We select each motif from the set and calculate number of his occurrences in the input dataset. The motif with the highest score is the best motif based on selected motif representation.

The algorithm is guaranteed to find the best motif. Another advantage is that it is relatively scalable, allowing to include several forms of motif degeneracy. In addition each occurrence can be scored separately and therefore we can determine more complex score scheme as the number of the motif occurrences. Assume that we have a model that allows a flexible number of gaps between two motif positions. The motif occurrences with more nucleotides between the positions will be rated by smaller score and therefore contribute less to total score as well.

The main drawback of the exhaustive search is the computational time complexity, that is approximately $O(NmA^eL^e)$, where N is the number of the input sequences, m is their length, A is the alphabet size, L is the motif length and e is the number of errors allowed in a possible match (13).

An example of an algorithm that uses the exhaustive search is the MOTIF algorithm (14). The algorithm is dedicated to motif discovery in protein sequences. A motif is triplet of amino acids separated by fixed number of gaps. The length of gap range from $0, 1, \ldots d$, where d is value provided by user. The highest possible value is 24.

1.3.2 Prunning enumeration

The search space of simple exhaustive methods grows exponentially with increase in the complexity of the motif model. The complete enumeration is thus possible only for simpler motifs. Several methods have been proposed to address this problem. Here we present one of the method for reduction of the search space during the motif discovery process.

Assume that we are searching for all ungapped motifs, that occurs in at least k positions in the input sequences. In the first place, we prepare motifs of some small length (for example 1) that are located in at least k positions. Each motif is then extended in each possible way. The extended samples, that do not occurs in at least k positions are discarded from further analysis. All the samples that are extensions of the discarded motif are automatically discarded from the analysis as well. This approach can still lead to exponential time (for example for $k = 0$), however it can be used to reduction of the search space.

This tree can be traversed by a breadth-first or a depth-first way. These methods have been widely studied in (15). The main advantage of the breadth-first approach is that we can use previously calculated values to discard motif without examining its quantity in the sequences.

Assume that we have already estimated quantity of each motif of length 2. The motifs, that did not have sufficient occurrences were discarded. Lets say that we discarded only one motif, **cg**. We can now discard all of the motifs of length 3 that contains this motif, that would be motifs **acg**, **ccg**, **gcg** and **tcg**.

Unfortunately this approach can be applied only for short motifs, because the number of the stored motifs in each level grows exponentially. Therefore the depth-first search is better choice for the space efficiency.

The example of the algorithms that use the prunning for reduction of the search space is the Pratt algorithm (16).

1.4 Gibbs sampling methods

In previous section we examined deterministic algorithms that use an simple word-based motif model. Described algorithms guaranteed to find the best motif. With a more complex probabilistic motif model we cannot hope to do so. The probabilistic models have parameters that are of continuous space and therefore it is not possible to enumerate all values. Therefore iterative algorithms are used, like EM, that will be described in detail in the next chapter, Baum-Welch, which is an application of HMM and Gibbs sampling described here.

The Gibbs sampling can be described as Markov Chain Monte Carlo (MCMC) process: "Markov-Chain", since the results from every step depends only on the results of the preceding one like in EM; "Monte-Carlo", since the way to select the next step is not deterministic but rather based on the random sampling (17).

A motif discovery algorithm based on the Gibbs sampling method has been proposed in (18). We will introduce it briefly and discuss its drawbacks and proposed solutions.

1.4.1 Basic algorithm

Assume we are discovering a motif represented as PWM, of fixed width w that has at least one occurrence in each of the input sequences $Y = (Y_1, Y_2 \ldots, Y_N)$. The algorithm is initialised by choosing random starting positions of motif occurrences. For each sequence i from dataset Y, one substring of length w is selected randomly. Positions of the samples are stored in set of positions $o = (o_1, o_2, \ldots, o_Y)$, where $o_i = p$ denotes motif occurrence on i-th sequence $M_i = (Y_p, Y_{p+1}, \ldots, Y_{p+w-1})$. Each stored sample is treated as motif occurrence and is used for calculation of PWM model using the equation

$$f_{ij} = \frac{c_{ij} + \beta_j}{\sum_{k=1}^{L} c_{ik} + B} \tag{1.1}$$

Value c_{ij} denotes number of observed occurrences of the nucleotide j on the k-th motif position. To avoid zero values in the matrix f, we will add positive constants, also known as pseudocounts, one for each nucleotide type, $\beta = (\beta_1, \beta_2, \ldots, \beta_L)$. Value B is calculated as the sum of pseudocounts, $B = \sum_{j=1}^{L} \beta_j$.

The algorithm is iteratively changing one of the motif positions. Modified solution does not have to be necessarily better than the previous solution, but changes leading to an improvement are chosen with a higher probability. This approach helps overcome local maximum.

The first step of each iteration is called the predictive step. One of the sequence (Y_i) from the dataset is chosen either randomly or in a specified order. A new motif model estimate is computed based on the motif instances located at $o_1, o_2, \ldots, o_N$, except the occurrence in the selected sequence, o_i.

New position of a motif instance in sequence i is then determined by the process called *sampling step.* Every substring of length w of sequence Y_i is a possible candidate for motif occurrence. Their quality is estimated in order to prioritise samples that are more similar to the motif occurrences located at other sequences. This can be done by comparison of probabilities that a sample X_j was generated from the PWM and the background distribution.

$$S_j = \frac{P(X_j|f)}{P(X_j|b)} \tag{1.2}$$

Background distribution is calculated by counting nucleotides in non-motif areas.

$$b_j = \frac{d_j + \beta_j}{\sum_{k=1}^{L} d_k + B} \tag{1.3}$$

where d_j is number of occurrences of nucleotide j in non-motif areas.

The higher the score, the more similar is sample X_j to motif instances and more differ from the background distribution. One of the possibilities is to choose the most similar sample as motif instance, however this *greedy* approach results in suboptimal solutions, because it lacks ability to get away from a local maximum. A new location of a motif is usually chosen with probability proportional to its quality, that is $S_j / \sum_k S_k$ for sample X_j. A position of the selected sample X_l becomes new o_l.

1.4.2 Avoiding local maxima

Several problems of Gibbs sampling have been addressed. Although this method is more resistant to the problem of local maxima than greedy methods that chooses the best solution in each iteration, the presence of local maxima still significantly limits its strength.

One of the proposed change is to start from several starting points and examine the quality of each found solution. However, the task of selecting an appropriate set of initial candidates is quite difficult. We will investigate selection of such a set for the EM algorithm in chapter 3, perhaps our techniques can be used here as well.

The method called simulated annealing have been proposed to reduce effect of the local maxima (19). The main change is in the update step. A new parameter T called temperature is introduced. Probability of a choosing substring of the sequence i as the new motif occurrence is not proportional only to the quality of the sample, but also to the temperature. The temperature is initialised with a high value that causes, that samples with different qualities have similar probabilities to become the new locations of the motif. The temperature is then gradually decreased. The lower value causes that samples better matching current model are chosen with a much higher probability.

The method of the simulated annealing has been successfully used for several computational problems (20) that have a problem with local maxima as well, however experiments show that this approach does not bring significant improvement for motif discovery problem (19).

An improvement has been proposed in (21). The proposed change in this algorithm called GibbsST is to adjust the temperature adaptively to the current score. By changing the temperature, the GibbsST adopts continuously by changing search methods from a fast greedy search to a more random search, reducing the possibility of being trapped in local maxima.

1.5 Hidden Markov model algorithms

This section covers basic principles behind training motif representation based on hidden Markov models (HMM). The process includes constructing an appropriate topology of the

HMM and later training its parameters.

1.5.1 Topology

Before process of parameter training HMM, proper topology of HMM has to be established, to reflect our belief about input sequences. The model is most often composed from two components, one for background regions and the other one for the motif instances.

The background component in many cases consists only one state that emits nucleotides with probabilities which reflect their frequencies in the input sequences. The more complex model can be created to incorporate additional information about sequences. For example, in chapter 4.2 we create a special background model for that intergenic regions of yeast mitochondria. These sequences are enriched in adenine and thymine but they also contains features called GC islands which are regions highly enriched in cytosine and guanine. The background component which captures this distribution is composed from two states, AT rich and CG rich. Their emission rates reflect the frequencies in the AT rich regions and GC islands, respectively. An HMM for this background component is depicted in figure 4.1

Another component which is responsible for generating motif instances is the motif component, which is described in more detail in the section 1.2.3. It is necessary to decide the motif width before training a HMM, because the number of match states responds to the length of generated motif instances. Also we need to decide, if we want to add the insert and deletion states and therefore capture the possibility of gaps in motif occurrences. An incorporation of them will cause significant increase in the complexity of the model, which may lead to overfitting the data.

In the figure 1.5 we present an example of the HMM topology. The background component is composed from one state. The motif component is composed from six states, $position1, \ldots, position6$ and respond to frequency matrix of transcription factor ZEB1 visualised as sequence logo in figure 1.2.

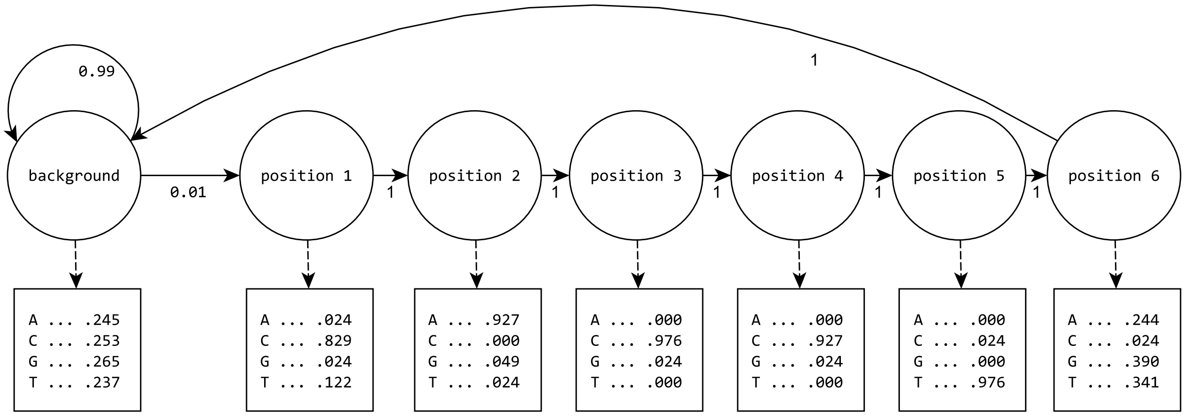

Figure 1.5: Hidden Markov model with motif component

1.5.2 Training

The next step of solving the motif discovery problem is estimate emission and transition probabilities of each state in the HMM, so that the model will generate similar sequences as those in the dataset. A frequently used algorithm for this task is the Baum-Welch algorithm.

The Baum-Welch algorithm is a particular case of the expectation-maximisation (EM) algorithm. We will discuss a case of the EM algorithm in chapter 2.3.

In the first step, initial parameters of the model are determined, either provided by the user, based on a prior information, or selected at random. Starting parameters are then iteratively improved. Probability of each path is first calculated by the *forward* and *background* algorithms (22) using the HMM parameters form the previous iteration. Then transitional and emission probabilities are estimated to maximise probabilities of these paths. After each iteration, a new set of HMM parameters is obtained that provide a better explanation of the input sequences. The algorithm eventually converges to a local maximum so good starting points are required to obtain reasonable results.

Chapter 2

MEME algorithm

MEME is one of the commonly used programs for motif discovery. The algorithm has been proposed by Elkan and Bailey in (23) and (24). We decided to base our work on this algorithm, customizing it for further tasks associated with specific structure of mitochondrial genomes due appropriate balance between its simplicity and accuracy. This chapter deals with its fundamental principles and implementation issues.

2.1 Model

2.1.1 Dataset

The MEME algorithm searches for maximum likehood estimates of the parameters of a finite mixture model which could have generated a given set of sequences. One component of the model describes motif occurrences, while the other component describes all other positions in the sequences. Fitting the model includes estimating parameters of both components as well as the relative frequency of motif occurrences.

MEME does not consider whole sequences, instead it examines all substrings of length w. To be more precise, assume, that we want to find a pattern of length w in sequences $Y = (Y_1, Y_2, \ldots, Y_N)$. MEME breaks up these sequences into n overlapping substrings of length w and thus creates a new dataset $X = (X_1, X_2, \ldots, X_n)$. The goal is to mark each sample either as a motif occurrence or random sequence from the background model.

Problem of this reduction is that it doesn't consider samples that contain a mixture of motif and background area. This is for example sequence, that contains motif occurrence, but it doesn't start at first position. Thus first nucleotides are generated by background model and the rest by motif model. Another problem is that samples from X dataset are not independent. However published results in (24) show that this relaxation is good approximation that simplifies problem without significant impact on discovery precision.

Each sample has assigned a vector of values that reflecting its membership to particular groups, in our case either motif or background component. All values are stored in matrix Z.

$$Z_{ij} = \begin{cases} 1 & \text{if } X_i \text{ was generated by } i\text{-th component} \\ 0 & \text{otherwise} \end{cases}$$

The membership of a i-th sample is normally ambiguous and thus Zij value is from interval $\langle 0, 1 \rangle$. The higher the Zij value, the better probability that i-th sample comes from the j-th component distribution.

Another unknown information is the relative number of samples generated by the individual components. It can be divided into quantity of the motif occurrences denoted as λ_1 and quantity of the background samples as $\lambda_2 = 1 - \lambda_1$.

$$\lambda = (\lambda_1, \lambda_2)$$

The main task of MEME algorithm is to estimate the parameters of motif and background component and compute Z values, so that they provide the best explanation for the sequence dataset and therefore achieve the best likehood score.

2.1.2 Position weight matrix

MEME algorithm is based on position weight matrix (PWM) motif model representation presented in detail in section 1.2.2. Formally, we define the motif as a $w \times L$ matrix f, where L denotes the number of distinct nucleotides. Each position in a sequence of length w which is an occurrence of the motif is generated as an independent random variable describing a multinomial trial with parameter $f_i = (f_{i1}, f_{i2}, \ldots, f_{iL})$. Entry

f_{ij} therefore stores the probability that nucleotide type indexed by j will be at the i-th position of motif occurrence.

Assuming column independence, probability, that sample X_i from dataset was generated by the motif component is defined as

$$\begin{aligned} P(X_i|f) &= \prod_{j=1}^{w} P(X_{ij}|f_j) && (2.1) \\ &= \prod_{j=1}^{w} \sum_{k=1}^{L} I_{ijk} f_{jk} && (2.2) \end{aligned}$$

where

$$I_{ijk} = \begin{cases} 1 & \text{if } X_{ij} = k \\ 0 & \text{otherwise} \end{cases}$$

MEME assumes that a w-length sequence which is not occurrence of the motif is a sequence of w nucleotides independently generated from a single background distribution. Based on this assumption, calculation of the background probability is

$$\begin{aligned} P(X_i|b) &= \prod_{j=1}^{w} P(X_{ij}|b) && (2.3) \\ &= \prod_{j=1}^{w} \sum_{k=1}^{L} I_{ijk} b_k && (2.4) \end{aligned}$$

Initial background distribution b can be obtained from the input sequences, by simple frequency counting

$$b_i = \frac{c_i}{\sum_{j=1}^{L} c_j} \qquad (2.5)$$

where c_i is the number of occurrences of the i-th residue type in the samples. This calculation considers all residues from sequences as part of the background, but due to relatively small expected proportion of motif instances, it can be used as good estimation of real frequencies.

In following sections we will use notation θ_1 and θ_2 as substitution for f and b, respectively.

$$\theta = (\theta_1, \theta_2)$$

2.2 Likehood of the estimated parameters

During the computation, we may obtain various estimates of unknown parameters. Therefore we need to establish general method for their comparison. The MEME algorithm uses calculation of likehood function which expresses, how well the estimated model parameters explain sequences in the dataset. The higher the likehood value, the better the explanation of the data, and therefore the better parameter estimates. MEME algorithm is trying to maximise likehood function to obtain best possible explanation for input sequences.

2.2.1 Calculation

The likehood of the model parameters θ and λ given the joint distribution of the data X and missing data Z is defined as

$$L(\theta, \lambda | X, Z) = p(X, Z | \theta, \lambda) \tag{2.6}$$

Probability of obtaining data X and Z from model characterized by parameters θ and λ is

$$p(X, Z | \theta, \lambda) = \prod_{i=1}^{n} p(X_i, Z_i | \theta, \lambda) \tag{2.7}$$

Using definition of conditional probability we can write

$$\begin{aligned}
p(X, Z|\theta, \lambda) &= \prod_{i=1}^{n} p(X_i|Z_i, \theta, \lambda) p(Z_i|\theta, \lambda) && (2.8)\\
&= \prod_{i=1}^{n} (\prod_{j=1}^{g} p(X_i|\theta_j)^{Z_{ij}} \prod_{j=1}^{g} \lambda_j^{Z_{ij}}) && (2.9)\\
&= \prod_{i=1}^{n} \prod_{j=1}^{g} (p(X_i|\theta_j)\lambda_j)^{Z_{ij}} && (2.10)
\end{aligned}$$

where g is the number of components in the model.

2.2.2 Computational issues

Calculation of the likehood is problematic due to numerical issues that arise from multiplying many numbers smaller than one. For example Java's most precise data type for storing real number values is Double. Smallest positive number that can be stored in this type is 2^{-1022}. Empirical results show that this limitation can be exhausted by evaluating roughly 100 samples.

In order to find the maximum of the likehood function, we can apply an increasing function defined on the codomain of L that is $\langle 0, 1 \rangle$. Good candidate for such function is log function. This will simplify calculations by replacing multiplications with additions and thus speed up calculation.

Using $\log L$ metric we obtain

$$\begin{aligned}
\log L(\theta, \lambda|X, Z) &= \log p(X, Z|\theta, \lambda) && (2.11)\\
&= \log \prod_{i=1}^{n} \prod_{j=1}^{g} (p(X_i|\theta_j)\lambda_j)^{Z_{ij}} && (2.12)\\
&= \sum_{i=1}^{n} \sum_{j=1}^{g} Z_{ij} \log(p(X_i|\theta_j)\lambda_j) && (2.13)\\
&= \sum_{i=1}^{n} \sum_{j=1}^{g} Z_{ij} \log p(X_i|\theta_j) + \sum_{i=1}^{n} \sum_{j=1}^{g} Z_{ij} \log \lambda_j && (2.14)
\end{aligned}$$

We will use this objective function to compare results from different runs of MEME algorithm and choose the best candidate for motif with highest log likehood value.

2.3 EM algorithm

In previous section we have introduced the scoring function (log *likehood*) which we use to express, how good are estimated model parameters. To provide the best explanation for input sequences, we need a method for finding parameters that maximises this function. For this purpose, the MEME algorithm uses expectation-maximisation (EM) algorithm (25).

Algorithm 1 EM algorithm

```
$\theta^{(0)}, \lambda^{(0)} \leftarrow$ initial model parameter values
repeat
    estimate samples membership $Z^{(n)}$ based on current estimates of $\theta^{(n)}, \lambda^{(n)}$
    estimate new values of $\theta^{(n+1)}, \lambda^{(n+1)}$ based on samples membership ($Z^{(n)}$)
until change is too small
```

EM is an iterative process. It switches between performing an expectation (E) step, which computes the expectation of the log-likelihood evaluated using the current estimate for the motif locations (Z), and a maximisation (M) step, which improves model by estimating parameters to maximise the log likelihood function based on previously founded motif occurrences.

This process is guaranteed to find better parameter estimates in each iteration until it converges. The main drawback of this method is that process finds only local maximum. Therefore good initial parameters close to the best solution are required to converge to the global maximum.

2.3.1 Expectation step

Missing values about locations of motif occurrences are calculated in the expectation step. New Z values are estimated, where Z_{ij} value express belief that sample X_i is a member of group represented by j-th component. We assume that samples come from model components, thus the expected Z_{ij} value is calculated as a proportion between probability that sample X_i was generated by j-th component and probability that was generated by any component from model.

$$
\begin{aligned}
Z_{ij} &= E[Z_{ij}|X_i,\theta,\lambda] && (2.15)\\
&= p(Z_{ij}=1|X_i,\theta,\lambda) && (2.16)\\
&= \frac{p(Z_{ij}=1\wedge X_i|\theta,\lambda)}{p(X_i|\theta,\lambda)} && (2.17)\\
&= \frac{p(Z_{ij}=1\wedge X_i|\theta,\lambda)}{\sum_{k=1}^{g} p(Z_{ik}=1\wedge X_i|\theta,\lambda)} && (2.18)\\
&= \frac{p(X_i|Z_{ij},\theta,\lambda)p(Z_{ij}=1|\theta,\lambda)}{\sum_{k=1}^{g} p(X_i|Z_{ik},\theta,\lambda)p(Z_{ik}=1|\theta,\lambda)} && (2.19)\\
&= \frac{p(X_i|\theta_j)\lambda_j}{\sum_{k=1}^{g} p(X_i|\theta_k)\lambda_k} && (2.20)
\end{aligned}
$$

2.3.2 Maximisation step

In the M-step we adjust model parameters θ and λ in order to find their better estimates. The maximisation of *log*-likehood function 2.14 over λ involves only second term.

$$\lambda^{(n+1)} = \arg\max_{\lambda} \sum_{i=1}^{n}\sum_{j=1}^{g} Z_{ij}^{(n)} \log \lambda_j \tag{2.21}$$

In other words we want estimate of λ_j which is relative quantity of samples that was generated by j-th component. We have already estimated membership of each sample and component (stored in Z matrix), so the best explanation for these values will be frequency of such samples in dataset. Thus we can calculate new values for λ as

$$\lambda_j^{(n+1)} = \sum_{i=1}^{n} \frac{Z_{ij}^{(n)}}{n} \tag{2.22}$$

Calculation of new estimate for θ will affect the value of the *log*-likehood function by changing the value of the first term in 2.14

$$\theta_j^{(n+1)} = \arg\max_{\theta_j} \sum_{i=1}^{n} Z_{ij}^{(n)} \log p(X_i|\theta_j) \tag{2.23}$$

Again, we want to calculate the best possible explanation of generating existing samples with already estimated membership to model components. For background component, it is just finding samples that were estimated as background and find frequency of each nucleotide in such samples.

$$b_k = \frac{c_k}{\sum_{l=1}^{L} c_l} \tag{2.24}$$

where

$$c_j = \sum_{i=1}^{n} \sum_{j=1}^{W} \sum_{k=1}^{L} Z_{i2} I_{ijk} \tag{2.25}$$

For motif model we need to estimate nucleotide frequencies for each column separately . Again, we collect samples that are annotated as instances of the motif. Then i-th nucleotide of each sample affects only f_j distribution, calculation is

$$f_{jk} = \frac{d_{jk}}{\sum_{l=1}^{L} d_{jl}} \tag{2.26}$$

where

$$d_{jk} = \sum_{i=1}^{n} \sum_{k=1}^{L} Z_{i1} I_{ijk} \tag{2.27}$$

Chapter 3

Improvement of the MEME algorithm

The EM algorithm is a very powerful method of solving numerous computational problems in probabilistic models with missing data. Its biggest drawback is its tendency to find only local maxima. It starts with initial values of parameters of interest, and it improves them iteratively. Process under some conditions guarantee to find a locally optimal solution, however with increasing complexity of problem, chance, that the local optimum is also the globally best solution to the problem significantly decreases.

The MEME algorithm uses adjusted EM algorithm as its core component, and thus suffers from the problem as well. Model parameters θ and λ form $((w+1)L+1)$-dimensional space and it is difficult to initialize them with values. Initial values of some parameters can be obtained from data (approximation of the background distribution described in section 2.1.2), some can be provided by the user (relative frequency of motif instances), however simple method for estimating appropriate motif model parameters is unclear. Uncertainty involves the width of the motif as well as the expected frequencies of individual nucleotides at each position of the motif.

We propose changes in estimation of the initial parameters that lead to better speed and accuracy of the MEME motif discovery process. Our approach is based on searching for nucleotide pairs which occur frequently in the input sequences. Next we examine their neighbourhood to estimate the number of motif occurrences in the dataset and initial distributions for each position of the motif.

In the following sections, some methods for initial model estimation are described with their advantages and drawbacks.

3.1 Solution space

One of the simplest possibilities for initializing the EM algorithm is to set the distribution at each position of the motif equal to the background distribution, however this approach will reduce the motif model to the background model. Empirical results confirmed that the algorithm with such a starting motif model will not be able to overcome a starting point and find an appropriate motif candidate.

Another option is to go through every possible motif candidate. With this approach, algorithm guaranties to find best solution, because the best solution lies in the same domain as all acceptable solutions and therefore will be eventually examined as one of the possible candidates. However due to continuity of the candidate space, the number of possible candidates is infinite. The problem can be relaxed by discretizing the space. For example, for each position in motif model we can choose one dominant nucleotide and assign it emission probability p that will be greater than emission probability of other nucleotides. This approach will generate 4^w candidates of width w. That means $\sim 10^6$ candidates for a motif of length 10. The problem becomes more unmanageable, when we realise, that we must consider each possible motif width w, which gives us the following number of candidates.

$$\sum_{w=w_{min}}^{w_{max}} 4^w \tag{3.1}$$

For better understanding of biologically relevant motif lengths we used Jaspar database (2). It contains collections of transcription factor binding sites for multi-cellular eukaryotes. Motifs have been obtained from published binding sites experimentally verified by *in vitro* protein-DNA interactions.

We collected all motifs from the site for further investigation and converted each of them into a frequency matrix f. Then we determined the shortest and the widest known motif to establish a lower and upper bound of w value. These bounds restrict motif width to values from 4 to 30 nucleotides.Based on equation 3.1, $\sim 10^{18}$ possible candidates would need to be examined to consider all values for $w \in \langle 4, 30 \rangle$.

This example shows that the search in whole candidate set is impossible due to its huge cardinality. In addition, we simplified our problem by assuming that only one nucleotide is dominant at each motif position. In fact, two or even three nucleotides can be observed more frequently at a given position than other, non-dominant nucleotides. If we take into account possibility this rise in the number of candidates will be significant.

Due to problems arising from searching in candidate set systematically, some subset must be selected for further examination.

3.2 MEME approach

The MEME algorithm selects initial model parameters based on the dataset. A substring of length w is selected randomly from the provided sequences. This sample is used to estimate distributions of nucleotide types for each motif position. Assuming the selected sample is $x = (x_1,\ x_2, \ldots, x_w)$, the frequency of j-th nucleotide type on the i-th position is initialised as

$$f_{ij} = \begin{cases} m & \text{if } x_i = a_j \\ \frac{1-m}{L-1} & \text{if } x_i \neq a_j \end{cases}$$

where L is number of nucleotides.

In other words, sample is treated a motif occurrence. Without additional information, observed nucleotides are considered to be more significant for functionality of the motif and therefore their probability at the corresponding position will be set to value m which must be greater a $1/L$.

Multiple samples have to be selected to improve the chance that a motif instance is chosen. As a result, we obtain multiple initial models for examination. It is not plausible to run the EM algorithm from each model, therefore some method is required for estimation of their quality.

The following heuristic has been proposed to estimate the goodness of each model. One expectation and maximisation step of the EM algorithm is executed to improve the initial model. The candidate that achieves the highest log likehood value after evaluating these steps is chosen for further EM convergence.

This approach has two main disadvantages. First of all, it needs a prior information about the searched motif width which is not available. This drawback can be obviated by multiple runs with changes in parameter w. However such approach slows down the algorithm by a factor of (w_{max} - w_{min}).

Another problem lies in the candidate selection. We are choosing candidates directly from data, so it is guaranteed that a motif occurrence is one of the possible substrings. Using of all possible substrings as candidates results in an algorithm with running time $O(n^2w^2)$ where n is number of overlapping subsequences in dataset and w is a examined motif width. This running time considers only the initialisation phase with one iteration of EM per candidate.

Fortunately, since we assume that the input sequences contain multiple instances of the motif, we do not have to choose all substrings, but only a certain fraction of them. In addition, number of required samples does not depend on the dataset size, only on the frequency of motif occurrences and the desired accuracy of the solution.

If we randomly select a sample X from dataset, the probability that it isn't an instance of the motif is

$$p(X_i \text{ is not instance}) = (1 - \lambda_1)$$

If we choose Q samples, the probability that none of them is an instance of the motif is

$$p(\text{no occurrence chosen}) = (1 - \lambda_1)^Q$$

To make the probability that we choose at least one occurrence higher than the defined threshold α, $0 \leq \alpha < 1$, we choose Q value, such as

$$(1 - \lambda_1)^Q \leq 1 - \alpha$$

which happens when

$$Q \geq \frac{\log(1-\alpha)}{\log(1-\lambda)} \tag{3.2}$$

More then Q samples are needed to examine in order to find an instance of the motif with the probability of a success higher then α. Notice, that the number of required samples is not changed with increase of the dataset size.

3.3 Selection based on probes

The approach used by the MEME algorithm for selecting initial model parameters suffers from several drawbacks resulting from its stochastic nature. We propose a deterministic solution for finding a starting motif model estimate based on searching for combinations of two nucleotides at a certain distance that occur more frequently than expected at random. It is probable that their increased frequency is caused by involvement in motif instances, and so they are good candidates for further examination.

An identification of significant pairs is first step for selecting initial model parameters. Assume a motif of width 6 and pair **C..G** with distance 3. There are 3 possible placements of the pair in the motif, **(C..G..)**, **(.C..G.)** and **(..C..G)**. The placement **(C..G..)** will set initial distribution of the first and fourth position of the motif so that cytosine and guanine, respectively, have significantly higher frequency on associated positions than other nucleotides. Proper placement of the pair is important for the further analysis.

We need to establish distributions of the other motif positions as well. This is done by an examination of the nucleotides in the neighbourhood of the pair locations. Another parameter is quantity of the motif occurrences, which is estimated by examining the difference between the observed and expected number of locations of the pair.

In this section we deal with a problem of searching for pairs of nucleotides with this property. Issues regarding estimation of initial model from these pairs are covered in the following sections.

3.3.1 Probe definition

The following notation will be useful. A probe is a pair of nucleotides with associated distance between them. For example $(A, C, 5)$ is a probe parametrised with adenine and cytosine, respectively, with distance 5. We say that probe (p, r, d) fits at position i in the sequence, if the nucleotide at position i is p and the nucleotide at position $i + d$, is r. The number of locations in dataset Y, where a probe fits the sequence is quantified by function

$$D_Y : L \times L \times N^+ \rightarrow \mathbb{N}_0$$

Assume a dataset Y that contains no motif instances. We expect that such a dataset is generated by a model composed only from the background state. Each position in the sequence is generated independently, by a multinomial trial random variable b. Based on the assumption of independence of each position, the expected number of location where probe (p, r, d) fits to sequence from dataset Y depends only on background emission probabilities of nucleotides p and r.

$$E[D_Y(p, r, d)] = b_p b_r \tag{3.3}$$

Notice that this quantity is the same for every value of d.

When we consider sequences with motif occurrences, D values will change due to different emission probabilities of the motif model. Changes would be more significant with higher frequency of motif instances and also with rising discrepancy between background and motif model emissions. To be more concrete, we introduce more terminology.

Nucleotide p is dominant at a particular motif position i, if it occurs more frequently at this position than other nucleotide types, formally $f_{ip} \geq f_{ir}, \forall r \in L$. Intuitivelly we would expect that the greater the difference between emission probability of the dominant and non-dominant nucleotides, the more significant is the dominant nucleotide for a proper functionality of the motif. We will say that a motif location i with dominant nucleotide p is the most dominant in motif if $f_{ip} \geq f_{jr}, \forall r \in L, j \in \{1..w\}$.

Our goal is to find a probe based on two most dominant motif positions and the distance between them. If we have multiple motif instances in the dataset, such a probe should be generated in many of them, thus the quantity of probe locations in the dataset would be higher than expected. These observations can be used to establish the initial model based on prior information obtained from data.

3.3.2 Quality of probe

More dominant probes change observed values of D function in the sense that their frequencies would be greater, than expected by chance. We introduce a new function R that will reflect these differences

$$R_Y(p,r,d) = \frac{D_Y(p,r,d)}{E[D_Y(p,r,d)]} \tag{3.4}$$

D function is determined directly from the provided dataset, counting the frequency of each probe. Expected value $E[D]$ is evaluated by estimating background emission b and substituting into equation 3.3. Background probabilities b for the calculation can be estimated by method mentioned in section 2.1.2. The higher difference between observed D and predicted $E[D]$ value, the better candidate for further examination.

3.3.3 Computational issues

Calculation of R value requires counting D and $E[D]$ value. Computation of $E[D]$ requires estimation of background probabilities b which can be approximated by method explained in section 2.1.2. Method requires to examine every nucleotide in sequences to determine quantity of each nucleotide type. From these counts it is straightforward to compute frequency with equation 2.24. Whole computation of $E[D]$ value can be done in $O(n)$.

D value is estimated from provided dataset as well. Process is similar as calculation of $E[D]$ value, but now pairs of nucleotides are recorded. Calculating quantity and latter frequency of probes can be done in $O(nV)$, assuming we set the maximum distance for probes, V.

We need to select probes with the highest R value that are the best candidates for further analysis. The simplest solution is to sort R values in the order of decreasing R values. Every probe is combination of two nucleotides and distance, so number of distinct

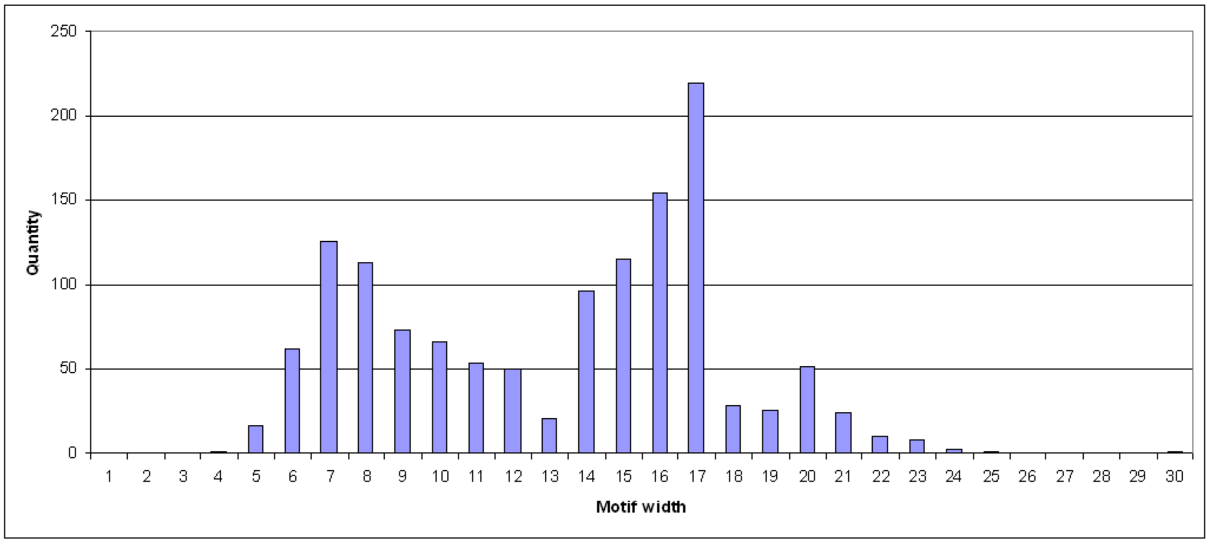

Figure 3.1: Number of motifs with different widths

probes are $|L||L|V$. The number of nucleotides is a constant for our problem, so sorting can be done in $O(V \log V)$ time. Complexity of this heuristic is thus $O(nV + V \log V)$.

3.3.4 Estimating maximum distance for probe

We used motifs from Jaspar database, to estimate adequate maximum distance between nucleotides in probe. One of the possibilities is to use width of longest motif. This value provides sufficient window for all motifs from the database, however a simple look at the motifs in the database indicates that we can use a smaller width. Only one motif in the database is of length 30. The second longest motif has width 25. Based on results shown in figure 3.1, window width 17 is sufficient for 88.6% of known motifs.

If we look closer at the known motifs, we observe that nucleotides located near the boundaries of a motif are usually less conserved and thus less probable to be dominant motif positions. To establish a more accurate maximum window width we investigated distances between two most dominant positions. Histogram of those distances is shown in figure 3.2.

If we choose window of size 10, we would cover most dominant positions in 99.24% published motifs. It is likely that even some of the remaining motifs contains some strong positions within this distance which will serve as good probes for further examination.

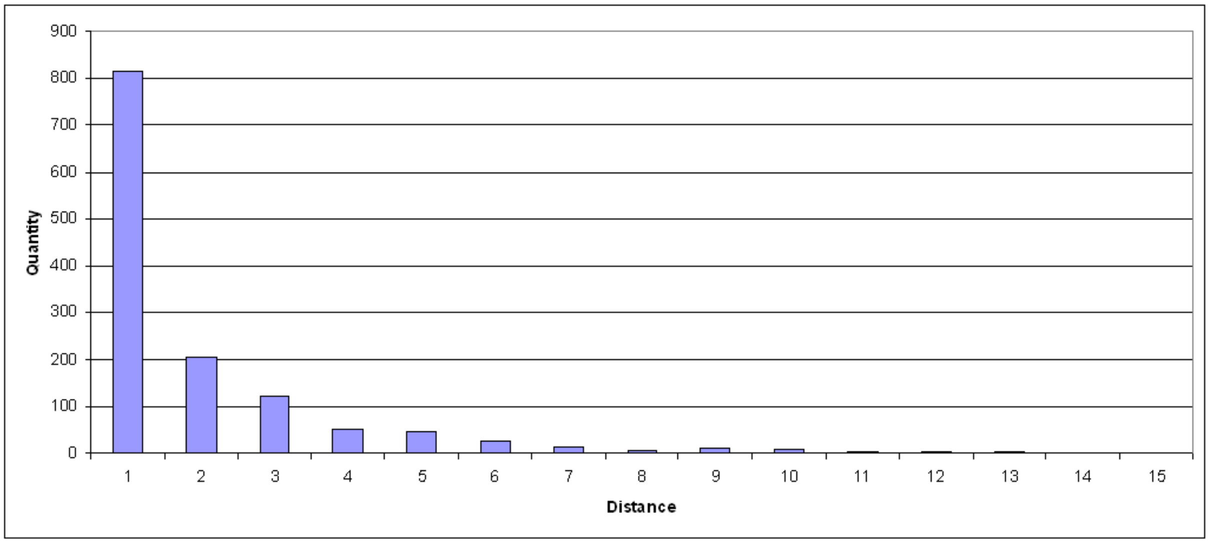

Figure 3.2: Distance between two most dominant positions in motifs.

Considering the small value of V we can consider it a constant which reduces the time complexity to $O(n)$. Note however that examination of probes is only the first step to estimate the initial motif model parameters.

3.3.5 Estimating the strength of a probe

This approach gives rise to the question if every motif has at least one strong probe or in other words two positions occupied by strongly dominant nucleotides. Again we used the published motifs from Jaspar database to determine the strength of dominant probes.

Every motif profile has been examined to localise two most dominant positions. We considered only the weaker of these positions which had lower frequency of dominant nucleotide. Histogram of the frequency (rounded to nearest integer) is displayed in figure 3.3.

The results support our hypothesis that motifs have at least 2 strong positions occupied by dominant nucleotides. From 1316 examined motifs 1219 $(92,63\%)$ have emission probability at least 0.9 for both dominant nucleotides. Such emission probability should be sufficient for significant variations in R values.

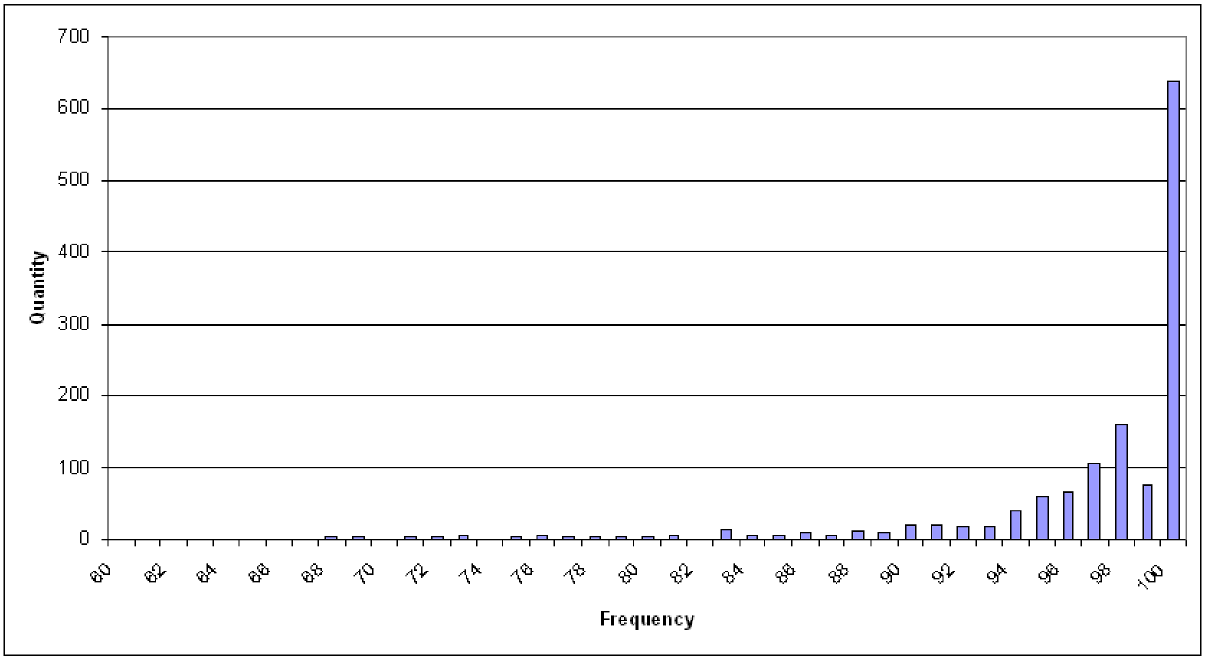

Figure 3.3: Emission probability of the second most dominant nucleotide

3.4 Probe discovery

Previous tests show that a majority of known motifs contain at least two nucleotides that can be used as a starting point for EM iteration. However, to explore power of this heuristic, it is necessary to examine, if the changes of R values are sufficient for selecting appropriate candidates for EM iterations.

3.4.1 Influence of dataset size

First, we created synthetic datasets with varying sizes to show evidence that an increase of samples leads to more precise D values and therefore to more accurate ordering of probes by R values.

Datasets were generated from a hidden Markov model shown in figure 3.4. Generation of sequences starts in the background state parametrised by emission probabilities b. We used the frequency of nucleotides in intergenic areas of yeasts as the background distribution (A: 0.359, C: 0.130, G: 0.139, T: 0,371).

With transition probability λ_1, the model can change its state to motif component

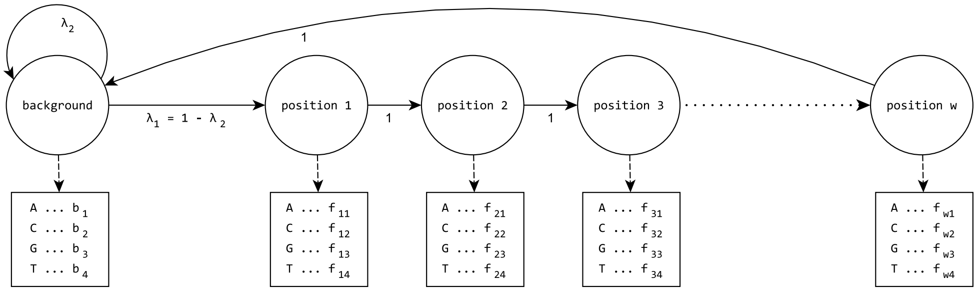

Figure 3.4: Full HMM with model and background component

composed from states *Position 1* ... *Position w*. We used value $\lambda_1 = 0.005$ in our experiments. State *Position i*, has emission distribution equal to f_i. After emitting exactly one nucleotide, responsibility for generation of next nucleotide is transfered to state *Position i+1*. After generating exactly w nucleotides which respond to a motif occurrence, the model continues to produce non-motif sequences by the background state. This process is repeated until sequences with desired lengths were created.

Motif model f was estimated from real motifs from the Jaspar database. A separate dataset was generated for each real motif.

First we wanted to see how increasing number of samples influences the accuracy of our solution. We executed our algorithm for finding R values on every created dataset. Then we order probes by R values. We were sequentially taking the highest R probes (and therefore the best candidates) and comparing them to the motif that was used in database creation. We were interested at position of first probe that fits to motif. We examined location of this value in each dataset. The histogram of such rank over all datasets is in the chart 3.5.

Result are not surprising, precision of our heuristic raise with increase in the sample count. If we consider a run as successful, when the correct probe has been situated in the first 10 positions, success rate of different dataset sizes is following.

- 500 samples: 87.31%
- 2500 samples: 89.89%
- 5000 samples: 93.69%

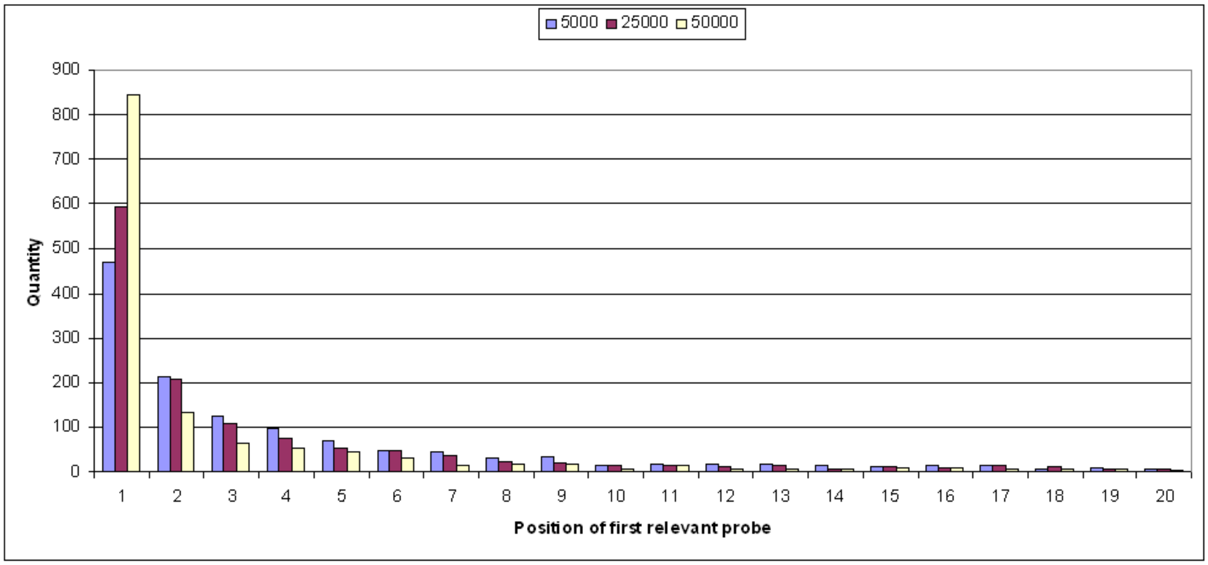

Figure 3.5: Rank of the first relevant probe with increase in dataset size

3.4.2 Influence of the quantity of motif occurrences

We have also examined, how the number of motif instances influences heuristic precision. Testing procedure was similar as in section 3.4.1. We created 3 datasets for each motif, each one with different motif occurrence probability (0.01, 0.005, 0.001). Each dataset contained 5000 samples. Evaluation was the same as in the previous test. We were searching for the first probe that fits to motif that was used for dataset creation. The results are displayed in figure 3.6

It is evident from the graph that precision of our heuristic significantly depends on the number of motif instances in the dataset. Success rate on these samples was the following:

- $\lambda_1 = 0.001$: 88.98%
- $\lambda_1 = 0.005$: 93.69%
- $\lambda_1 = 0.010$: 96.81%

3.4.3 Influence of the motif strength

In last test we were trying to decide, if the strength of most dominant nucleotides in motif has significant impact on heuristic accuracy. Hypothesis was tested on datasets with 5000

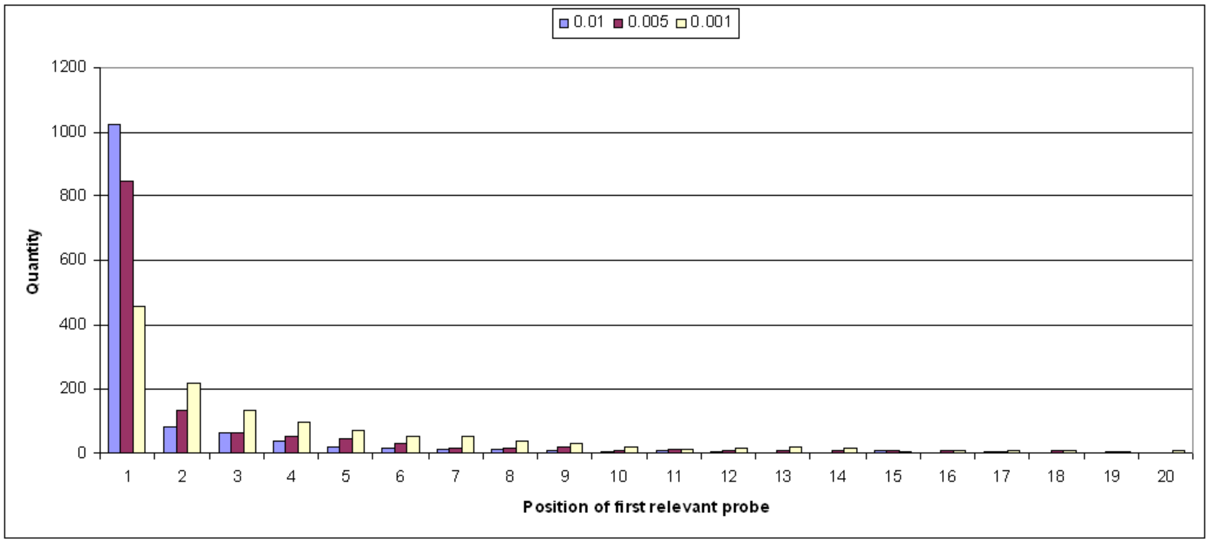

Figure 3.6: Position of first relevant probe with increase in motif occurrences

samples with motif occurrence rate $\lambda_1 = 0.001$. We assigned each motif value denoted as its strength which was equal to the frequency of the its second most dominant nucleotide. Again, we estimated the rank of the first relevant probe for each dataset. After evaluation, we calculated the average strength of motif for each rank. Results are displayed on the chart 3.7

Results do not show any significant decrease due to worse motif strength. However as shown in graph 3.3, the number of motifs with decreasing strength decline significantly and thus we have only few weak samples. Therefore more weak motifs are necessary to definitely reject this hypothesis.

3.5 Frequency of motif instances

In the previous section, we presented a method for estimating dominant positions in a motif with corresponding nucleotides and distance between them. Outcome of our algorithm is an array of probes ordered by decreasing relevance. We expect that in most cases, the higher the probe is ranked, the better its chance to fit to motif located in dataset. However a probe does not contain complete information about the motif model and therefore we need to collect more information to establish initial motif model parameters, specifi-

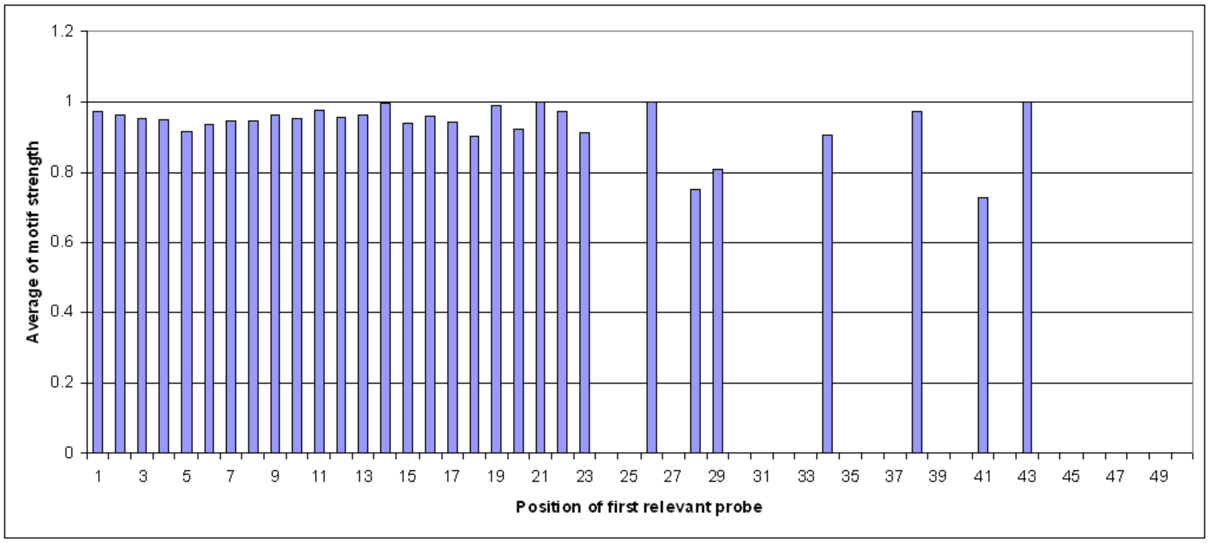

Figure 3.7: Average strength of motif for each position

cally motif width, place, where the probe fits to motif and emission probabilities of other positions in motif. In this section we propose heuristics for estimation of these parameters.

Significant factor that slows down the whole process of motif discovery is estimation of the initial λ parameter, which corresponds to the expected relative frequency of motif instances in the dataset. Without prior information, numerous values can be chosen and examined by the algorithm, again slowing down the computation. This method is used by the MEME algorithm. In this section we propose a method for estimation of λ parameter based on the discrepancy between the number of expected and observed locations of the probe.

3.5.1 MEME approach

The MEME algorithm selects λ values in following fashion. Assuming that at least $\sqrt{N}$ motif instances have to be present in dataset, the lowest considered value is set to $\frac{\sqrt{N}}{n}$. The highest value is estimated from the motif width as $\frac{1}{2W}$ to capture situation where one half of the non-overlapping substrings of length W are motif instances. Values of λ are chosen in a geometrically increasing series because experiments showed that starting points where λ was within a factor of 2 of the correct value were usually sufficient for EM to converge.

To illustrate slowdown caused by this approach, assume a dataset with 5 sequences of length 1000 in which we are trying to find motif of length 10. The algorithm would then consider 7 possible λ values, so the total time of initialisation will increase by a factor of 7.

Due to geometrical nature of selecting λ values, the number of required runs grows very slowly with increase in dataset size or decrease in motif width and it should not exceed 10 for common problem input. However using prior information from relevant probes we are able to estimate this value from sequences, and therefore decrease time required for computation.

3.5.2 Estimation based on probe

Assume that we have already estimated probes with associated R values. These values quantify the excess of probe occurrences observed in the data corresponded to the expected value computed from the common background distribution. We assume that sequence is generated by only two components, background and motif. Therefore the number of observed probe instances can be divide between these to models as

$$O(p, r, d) = O_b(p, r, d) + O_f(p, r, d) \tag{3.5}$$

where $O(p, r, d)$ is the total number of occurrences of the probe in dataset, divided into instances of motif ($O_f(p, r, d)$) and random occurrences ($O_b(p, r, d)$) in background sequence. Each probe location belongs to one of these groups. We can calculate the number of observed probe instances in $O(n)$ and also calculate the expected number of background occurrences of the probe as

$$E[O_b(p, r, d)] = nb_pb_r \tag{3.6}$$

This calculation can be also done in $O(n)$, which is time necessary to estimate the background distribution. The number of motif occurrences should respond to a difference between the number of observed and expected probe instances, $O(p, r, d) - E[O_b(p, r, d)$, and therefore the relative motif quantity is estimated as

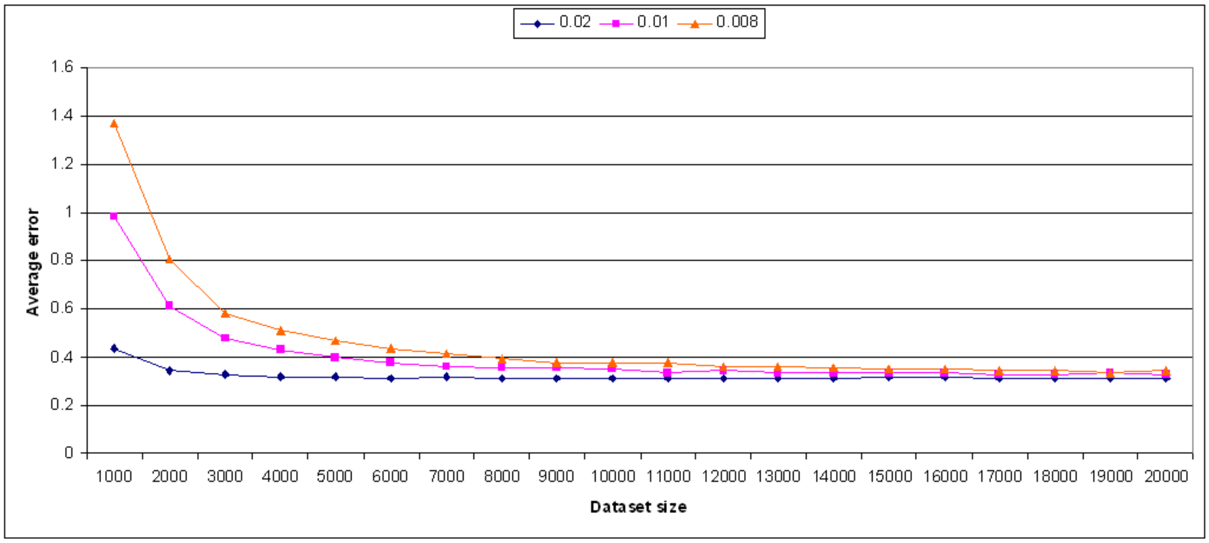

Figure 3.8: Error in estimating frequency of motif instances

$$\lambda_1 = \frac{O(p,r,d) - E[O_b(p,r,d)]}{n} \tag{3.7}$$

3.5.3 Experiment

We have executed a test to determine how precise is this heuristic. We created datasets for each motif from Jaspar database. Datasets differ in motif frequency, gradually 0.008, 0.01 and 0.02. For each motif multiple datasets were created with increasing size from 1000 to 20000 samples. The relative frequency of motif occurrences has been counted for each dataset and compared with the known value. Error was defined as ratio between the discrepancy and the correct value as

$$E = \frac{|\lambda_C - \lambda_R|}{\lambda_R}$$

where the λ_R is real value used for dataset creation and λ_C is the estimated value. Results are displayed in figure 3.8.

It is apparent that our heuristic requires a high number of samples, however in such a case it provides good estimates of relative frequency of motif instances. In majority of experiments the average relative error was below 50 %. For better accuracy, 3 initial values can be examined, namely the original estimate λ, and its two neighbors in the geometrical series, $\frac{\lambda_C}{2}$ and $2\lambda_C$.

3.6 Initial motif model

Assume that we have found a probe which is expected to be part of a motif of length w. This knowledge itself is not sufficient to form an initial motif model. We lack information about the probe position in motif and emission probabilities of other non-probe locations. In this section we present methods for estimating of these values.

3.6.1 Methods

Assume that we are searching for a motif of length w and we know a probe which fits to this motif. However we do not have any prior information about probe location within the motif, so additional computation is required. There are $w - d$ possible places in a motif of width w where we can place a probe with two given border nucleotides and $d - 1$ nucleotides between them. For more accurate estimate of its position we use EM steps to obtain likehood estimates of each position.

We were trying to measure accuracy of this heuristic simulating on similar datasets that have been used in section 3.4.1. Again we created a dataset for each motif from Jaspar database. Each dataset was created from 5 sequences of length 1000 to achieve $\sim$ 5000 motif candidates. Motif instances were generated with probability 0.01. From these datasets we obtained a list of ranked probes. We selected the first probe that fit to the motif and used it for further analysis. In real deployment, all skipped probes will be examined. Our experiment was concentrated on EM steps examination without influence of failures in probe searching.

We used the real length of the motif as w. This is another simplification, however the problem of selecting the correct width can be solved by multiple runs of the algorithm with change in w parameter. Our main goal was to examine, if such heuristic can correctly find the appropriate place for the probe in the initial motif model.

Multiple initial motif models were created, one for each possible position of the probe

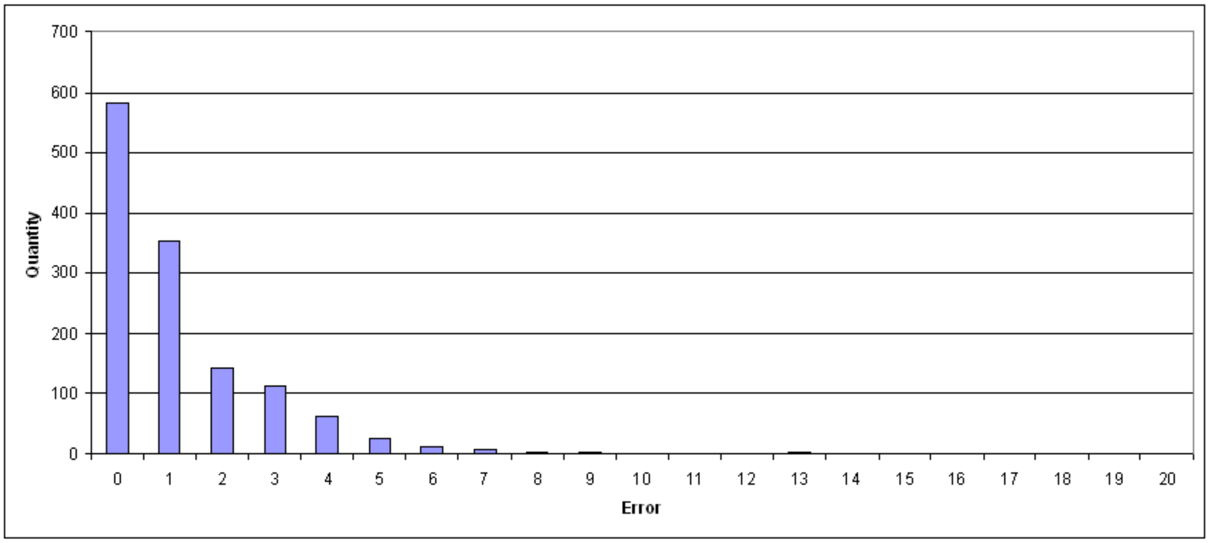

Figure 3.9: Error in estimating location of probe.

within the motif. Two dominant positions in motif marked by the probe had emission of the associated nucleotide set to value 0.9. Other positions had the same distribution as the background distribution estimated from the frequency of each nucleotide type in dataset. Each prepared motif served as an initial starting point for one EM iteration. After execution, log likehood function was computed, and an initial motif with highest score was selected as the best candidate for further EM computations.

3.6.2 Estimation based on probe

We examined position of the probe in this estimated initial model and compared it to the position of the probe in the real motif. The error function was defined as the difference between these two locations. Results are shown on chart 3.9

Results shows that only 29% positions have been determined correctly. This is only a small fraction, but fortunately the error in the remaining datasets was on average quite small(2.4) compared to the average length of the motif in the Jaspar database. Moreover approximately half of samples (50.22%) was shifted by at most one. Thus it seems that this heuristic can obtain reasonable results.

We have examined if multiple runs of the EM algorithm would not improve the accuracy of this heuristic. We executed the test once more with two and three EM iterations for

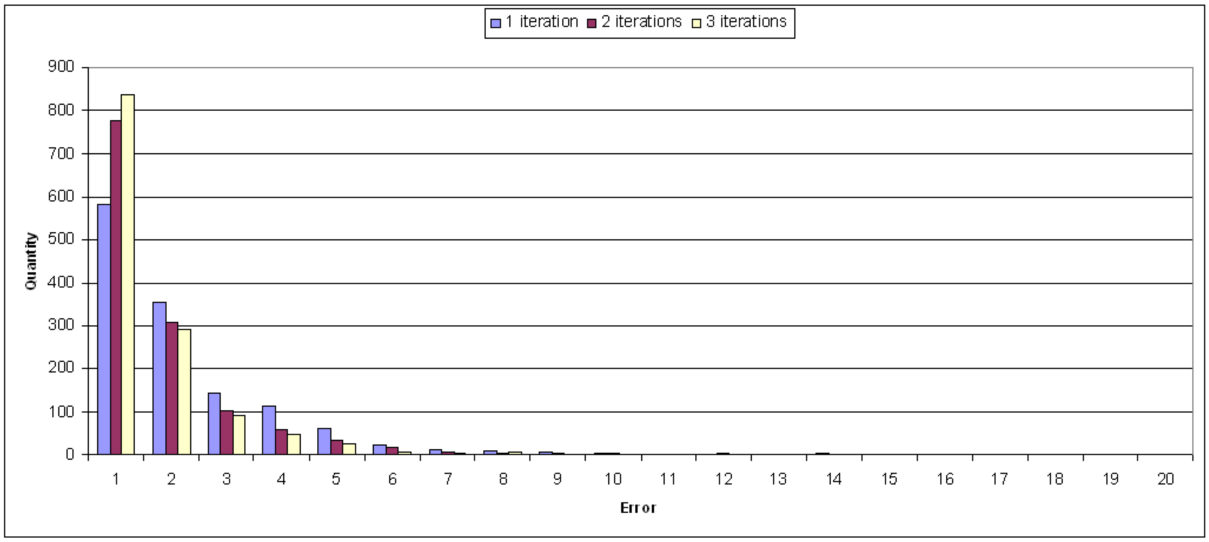

Figure 3.10: Impact of multiple EM iterations on heuristic

each motif candidate. The results provide evidence that multiple runs of the EM algorithm have positive impact on the algorithm precision (figure 3.6.2), however each iteration slows down the initialization phase of our algorithm.

One of the possible explanation for erroneous results is that displacement of the prove had been caused by weak nucleotides near the end of the motif. By weak we mean that their nucleotide emissions were too similar to background frequencies. To investigate this assumption, we classify positions in motifs on correctly placed and misplaced. Misplaced positions are those positions of the motif that are not included in the estimated motif. For each position we calculated the relative entropy with respect to the background distribution b, also known as Kullback-Leibler distance (26), which is defined as

$$dist_i = \sum_{j=1}^{L} f_{ij} \log_2 \frac{f_{ij}}{b_j} \tag{3.8}$$

Average distance for misplaced values was 0.594 and for correctly placed it was 1.1432. This result confirms our hypothesis that failures in the algorithm can be partially caused by positions that have emission probabilities similar to the background distribution.

3.6.3 Examination of probe surroundings

One of the problems of our heuristics was that non-probe motif positions were set to the background distribution. With our prior knowledge of two dominant nucleotides and the distance between them, we can estimate emission probabilities of individual nucleotide types in nearby positions and therefore enhance the initial starting point for EM iterations. Such improvement should lead to better reliability of $\log likehood$ function after one iteration along with more accurate estimate of probe location.

Again, we considered only the probe with highest R value that fits the real motif. There are $w - d$ possible places in the motif of width w where we can put two nucleotides with $d - 1$ nucleotides between them, so $w - d$ candidates for the motif were generated. Each candidate had associated a unique parameter $k = 0..w - d - 1$ which defines, how many nucleotides are placed before the probe. The number of nucleotides located after the probe is then $l = w - d - k - 1$. Every candidate is thus characterized by the pair (k, l).

We explored sequences in the dataset to find every occurrence of the probe. For motif candidate characterised by (k, l) we estimated nucleotide frequencies k positions before probe, l positions after probe and d positions between the two probes nucleotides. These frequencies have been used as emission probabilities for the motif candidate. With proper implementation, estimation of all possible candidates for probe location can be done in $O(wn)$.

We executed same set of tests as in section 3.6.1 to measure the benefit arising from this approach. Error had been counted in similar fashion, as the difference between the known and calculated position of the probe. Comparison of both methods is shown in figure 3.11.

Comparison shows that integration of the prior information about non-probe location can significantly enhance the accuracy of probe location estimation. In this heuristic, almost 60.3% of probes were correctly placed. The average error of our solution was 0.673, that is less then one misplaced location.

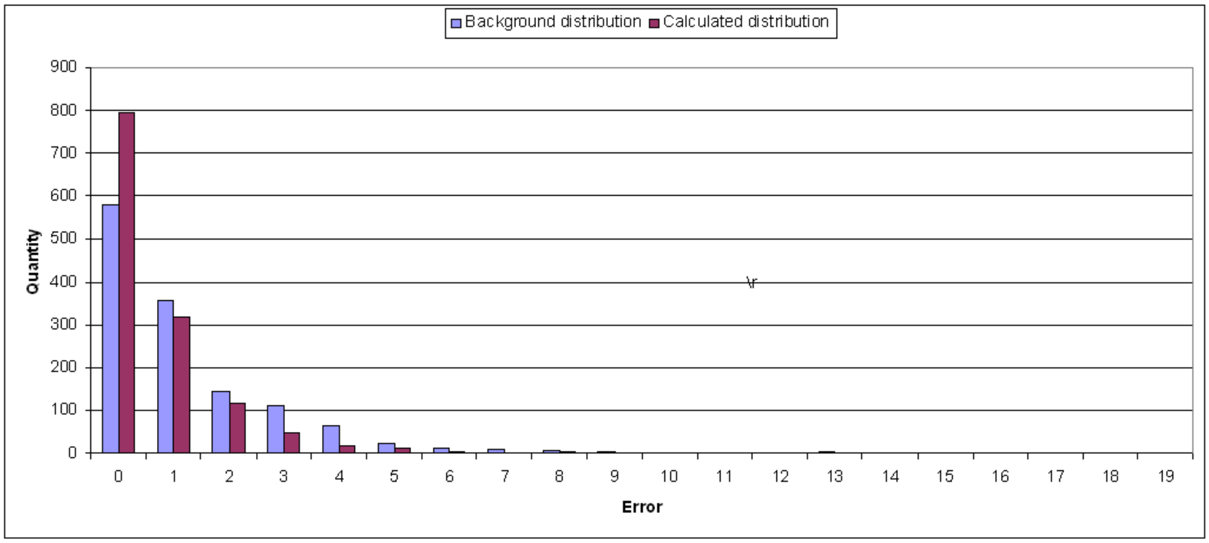

Figure 3.11: Comparison of using background versus estimated emission probability for non-probe locations

3.6.4 Improvement based on $\hat{c}$-values

In the heuristic proposed in the previous section, the surroundings of the probe were examined to estimate emission probabilities for each considered position. However such distributions are significantly affected by counting nucleotides that are not part of the motif occurrence. It happens when the probe is not part of an motif instance, but a pair of nucleotides from background sequence which occur randomly. In the initialization phase we do not have any information about location of the motifs. Fortunately, we are able to estimate the number of background occurrences, and therefore determine more precise initial distributions.

Again, we counted the number of occurrences of each nucleotide type at each position in a w-length window around the probe. We denote the number of nucleotide r at the i-th position of motif as c_{ir}. The number of probe occurrences is $C = \sum_{p=1}^{L} c_{ip}$ for any $i = 1..w$.

The expected number of non-relevant occurrences of nucleotide with background esmission probability b at position i is

$$E[c'_{ip}] = \lambda_2 b_p C$$

where $\lambda_2 = 1 - \lambda_1$ is the estimated frequency of background samples.

Methods for estimating relative frequency of motif instances have already been described in section 3.5. Now we can calculate the expected count of nucleotides of each type and position which are not instances of motif and eliminated them from further analysis. Obtained values can be used to gather better estimates for motif distributions $\tilde{c}$.

$$\tilde{c}_{ip} = c_{ip} - E[c'_{ip}]$$

Unfortunately, this approach can lead to negative nucleotide counts which is not plausible for distribution estimation, so a modification is required. The proposed change is to find the part that has highest negative impact on $\tilde{c}$ values. Such value can be understood as an error in λ estimation. Thus we search for ratio ω such that

$$\omega = \min((\min_{i,p} \frac{c_{ip}}{E[c'_{ip}]}), 1)$$

After multiplying values $E[c']$ with this ratio, it is highly probable that we obtain zero occurrence of some c_{ip} value. It is not plausible, because EM algorithm is not capable of modify such a value and therefore does not allow placement of nucleotide p on i-th position. To avoid this situation, some small value also known as pseudocount (β) has to be added to each count to express our uncertainty about the motif. The higher the number of samples and therefore lower uncertainty about data we obtained, the lower impact would such value have on motif emissions. Finally we can estimate the initial motif model as

$$\hat{c}_{ip} = (c_{ip} - \omega E[c'_{ip}]) + \beta$$

We computed $\hat{c}$ values for each possible placement of probe in motif. Then we used this distribution as the initial motif model, thus $f = \hat{c}$. With this parameter we executed one E and one M step of EM algorithm and calculated log likehood value of this proposition. Error was calculated in same way as in the previous tests, as the difference between the known and estimated position of the probe in a w-sized window. Comparison of values obtained from c-based and more advanced $\hat{c}$-based heuristic is shown in figure 3.12.

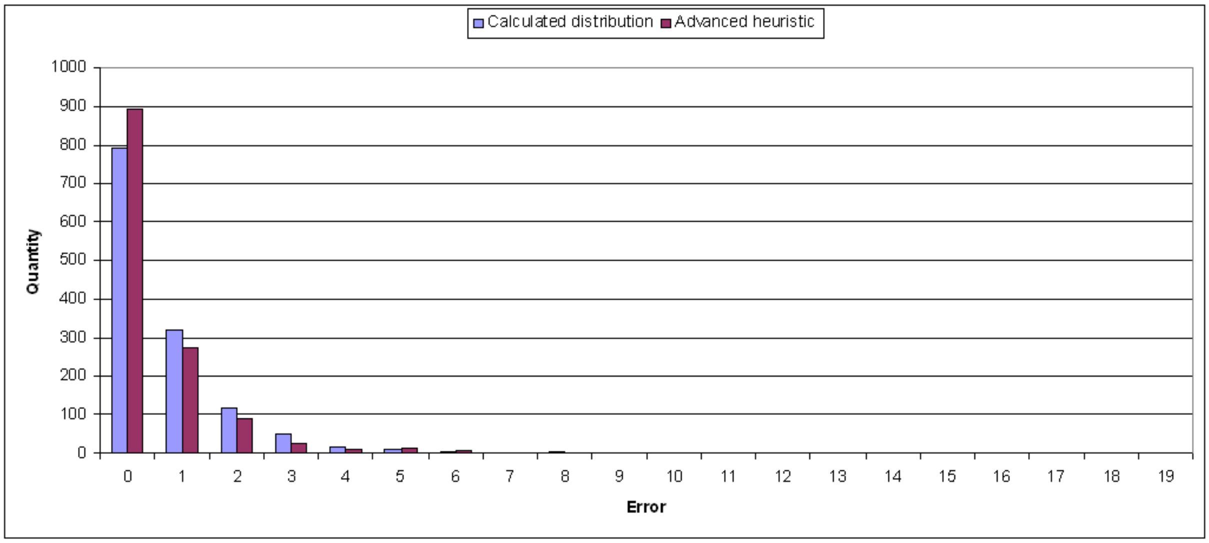

Figure 3.12: Comparison of c-based versus $\hat{c}$-based distribution

After subtraction the background distribution, we get more accurate probe location with probability of choosing correct position of 67,98% and average error 0.52.

3.6.5 Comparison of methods

Comparison of several proposed heuristics proposed in previous sections is summarised in table 3.1. Results obtained from estimated distributions of non-probe locations achieved significantly better results than using background distribution as default. Experiments based on c-based distributions achieved similar precision as three EM steps on b-based values. Considering that calculation of c-values can be quickly done in $O(n)$ and therefore does not bring significant increase in computational time, this heuristic can serve for more accurate probe location estimation.

The best accuracy was achieved by $\hat{c}$-values heuristic, with more precise estimation of initial motif model distribution. Contribution to computational time compared to calculation of c-values is negligible and therefore it should be used instead.

Experiments indicate that heuristics based on probe location can be very powerful method for deterministic estimation of initial model parameters for EM iterations.

Non-probe positions	Number of EM iterations	Correct estimates	Average error
Background	1	44.18%	1.23
Background	2	59.16%	0.8
Background	3	63.72%	0.64
c-values	1	60.3 %	0.67
$\hat{c}$-values	1	67.98%	0.52

Table 3.1: Comparison of MM algorithm with our heuristic

3.6.6 Impact on EM convergence

With estimated $\hat{c}$ values we are now able to initialise the EM algorithm with more accurate parameter values then in the case, when only the probe is known. Such improvement should lead to faster convergence of the EM algorithm.

To test this hypothesis we prepared a dataset for each motif with $\sim$ 5000 samples and motif probability 0.01. Then we find the highest ranking probe that fits to the real motif. We prepared two motif candidates for this probe that differ in distribution of non-probe location, one uses the background and other the estimated distribution. Again, we simplified the problem by choosing the correct probe and its location from the motif, to eliminate the influence of errors caused by estimating these values. We executed the whole EM algorithm until convergence with both motifs and record the number of performed EM steps and compared them.

EM algorithm with background distribution requires on average 94 iterations to achieve convergence. Model estimated by our heuristic with $\hat{c}$-values reached the convergence after approximately 77 runs. Improvement of speed was thus 23%.

The results demonstrate that the EM algorithm with the initial motif model based on probe discovery can be significantly accelerated by more detailed estimation of nucleotide occurrences around the probe locations present in the dataset.

Chapter 4

Motifs in mitochondrial DNA of yeast

Sequencing of mitochondrial genomes of multiple yeast species reveals their quite interesting structure. Intergenic regions are composed mostly from adenine (A) and thymine (T) nucleotides. These AT-rich regions contain nucleotide clusters highly enriched by cytosine (C) and guanine (G) (27) also known as GC clusters or GC islands. The more GC nucleotides these elements contain, the longer and more AT-rich are the intergenes containing them, leading to a direct relationship between the number of G and C nucleotides within the elements and the size of the genomes (28). In addition, sequence analyses indicate that the GC clusters are located mostly in the middle of the AT-rich areas (29). These observations lead to the assumption that they may fulfil a stabilising role for the AT rich areas (30).

We have adjusted the MEME algorithm to search for motifs in GC-rich areas. Discovery of motifs in GC islands may eventually divide them into families based on structural similarities and thus reveal shared evolutionary origin of GC islands from the same family.

4.1 Dataset

We are interested in GC islands which are located solely in intergenic areas of yeasts mitochondrial genomes. Because the genomes contain also functional elements, they had to be identified and filtered out to obtain purely intergenic regions. Another task was to localise GC clusters within intergenic regions. These two parts have already been performed by Juraj Mestanek and Peter Peresini in order to investigate changes in the number of GC clusters during evolution (31). In this section we describe data from this study which we will use for our experiments.

4.1.1 Separation of intergenic regions

We have analysed mitochondrial genomes from these yeast species: Candida albicans, Candida dubliniensis, Candida jiufengensis, Candida maltosa, Candida metapsilosis, Candida neerlandica, Candida orthopsilosis, Candida parapsilosis, Candida sojae, Candida tropicalis, Candida vartiovaarae, Candida viswanathii, Debaryomyces hansenii, Kluyveromyces lactis, Lodderomyces elongisporus, Pichia canadensis a Pichia farinosa.

Selected genomes contain non-coding regions as well as functional elements, so first task was to separate intergenic areas. Information about locations of functional units have been obtained from the GenBank database (32). Three types of functional regions have been considered:

1. protein coding genes
2. tRNA coding genes
3. rRNA coding genes

For further analysis we have used the regions between these functional elements as well as regions localised inside protein coding genes called introns. An intron is a segment of a gene situated between exons that is removed before translation of messenger RNA. It does not function in coding for protein synthesis and thus we have not considered it as a functional element.

4.1.2 Localisation of GC islands

A discovery of GC islands have been executed on the intergenic areas. The *minlength-cover* algorithm has been used to localise them (33). The detailed description of the discovery of the GC clusters is in (31).

Information about discovered GC clusters are shown in table 4.1. Species are sorted according to the number of the discovered GC clusters. We have used these GC clusters as a starting point in our analysis but we have reannoted intergenic sequences with a HMM described in the next section.

Species	GC clusters	Average cluster length	GC content of clusters
Candida tropicalis	120	46	63.71 %
Kluyveromyces lactis	114	43	67.87 %
Candida maltosa	96	33	62.64 %
Lodderomyces elongisporus	87	33	69.19 %
Candida albicans	85	46	62.34 %
Candida dubliniensis	65	43	62.54 %
Candida viswanathii	45	30	71.63 %
Candida sojae	44	30	76.81 %
Candida jiufengensis	27	35	65.68 %
Candida vartiovaarae	12	23	68.07 %
Debaryomyces hansenii	11	24	69.37 %
Candida metapsilosis	4	40	66.67 %
Candida parapsilosis	4	29	74.14 %
Candida orthopsilosis	1	35	82.86 %
Candida neerlandica	0	-	-
Pichia canadensis	0	-	-
Pichia farinosa	0	-	-

Table 4.1: Comparison of GC clusters of different yeast species

4.2 Hidden Markov model of GC clusters

Our goal is to use the positions of the GC clusters have been used as a prior information for a motif discovery process. We want to find motifs in the GC rich regions, thus their relevance should be higher than the relevance of the AT-rich regions. On the other hand, we do not want to completely exclude the AT-rich areas from our analysis in case that the GC cluster annotation does not agree with natural motif boundaries. A further annotation is thus required to assign each sample a value between 0 and 1, that would represent its membership to a GC cluster.

First, we establish parameters of the hidden Markov model, which would generate similar sequences as intergenic regions of mitochonrial DNA. The model has been used for estimation of a probability for each position in the sequences that it belongs to GC cluster. An appropriate combination of the values of a consecutive positions that form a sample can be used to express the sample relevance.

4.2.1 Estimation of model parameters

First step in construction of a HMM is establishment of its topology. We propose a simple two state model. One state (AT rich) is responsible for generation sequences similar to AT rich regions, another (CG rich) refers to GC clusters. Emission probabilities of both states are established by calculating frequencies of the nucleotides in the corresponding regions of the input sequences according to the existing GC cluster annotation. Similarly we estimate transition probabilities from the annotated GC clusters.

Estimated model with the emission and transition probabilities is shown in the figure 4.1.

4.2.2 Annotation of samples

Based on estimated model of intergenic areas, we are able to determine if it is more likely that a position in sequences belongs to a GC cluster or an AT rich area. We have used the *forward-backward* algorithm (34) for this annotation. The algorithm gets observed sequences and a hidden Markov model and for each position and state determines a probability, that the position has been generated by the state.

Our intention is to aim a motif discovery process to GC clusters. The relevance of Y_i nucleotide has been determined as the calculated probability that it has been generated by GC rich state. However, the MEME algorithm does not consider positions standalone,

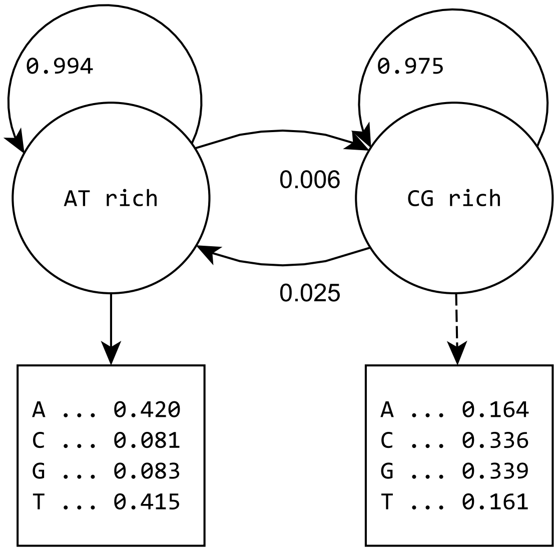

Figure 4.1: Model of intergenic areas of yeasts mitochodrial genome

instead it works with samples of w consecutive positions. Thus we need to determine a relevance of a sample as a combination of its positions. For this purpose, we have used the average relevance of the positions as the sample relevance. Other possibilities include median value or the relevance of the position in the middle of the sample.

4.3 Changes in the MEME algorithm

We have adjusted the MEME algorithm to reflect a specific structure of the intergenic areas of mitochodrial genome of yeast. This task involves expansion of the model, so that it is able to express AT rich regions as well as the GC clusters. Then we have used previously determined locations of the GC clusters as prior information of the motif discovery.

4.3.1 Changes in the model

The mixture model of the published version of the MEME algorithm (24) consists from two components, one describing background areas and one motif occurrences. This model is not appropriate for specific structure of mitochondrial genomes. The problem is with the background model which is unable to differentiate AT and the GC rich areas together which differ dramatically in their nucleotide frequencies.

We propose a solution of the problem by dividing the background model to two new components. The first background component, θ_2 express parameters of AT rich regions and the second one, θ_3, of GC clusters. Initial distributions have been obtained from the input sequences by calculating frequencies of the nucleotides in the corresponding regions. We have obtained these values:

$$\theta_2 = (0.420, 0.081, 0.083, 0.415) \tag{4.1}$$

$$\theta_3 = (0.164, 0.336, 0.339, 0.161) \tag{4.2}$$

Our model, $\theta = (\theta_1, \theta_2, \theta_3)$ is composed from three components, motif, AT rich and GC cluster component, respectively. For each component we need to estimate initial relative frequency $\lambda = (\lambda_1, \lambda_2, \lambda_3)$. First we establish λ_2 and λ_3 values, so that their proportion corresponds to the ratio between the number of the AT rich and GC cluster positions and $\lambda_2 + \lambda_3 = 1$. For our input sequences we have used values:

$$\lambda_2 = 0.807 \tag{4.3}$$

$$\lambda_3 = 0.193 \tag{4.4}$$

We are aiming the motif discovery process into GC rich areas, thus occurrences of a discovered motif should be situated there. The λ_3 value is therefore decreased by the provided λ_1 value, to ensure that $\sum_{i=1}^{3} \lambda_i = 1$.

4.3.2 Integration of the prior information

Our intention is to find motifs primarily in the GC clusters. We have incorporated prior information of the relevance of each sample obtained from the input sequences. Our approach for using relevance within motif discovery algorithm is similar to method how the MEME algorithm deals with a discovery of multiple motifs, published in (23).

Assume an input dataset that contains more more than one distinct motif. Each run of the MEME algorithm should localise the same, the most conserved motif. To discover a different motif in each subsequent run, the MEME algorithm erases the shared motif found by the EM and then repeats the EM algorithm to find the next shared motif. By removing each found motif, the MEME algorithm is able to find the next motif without interference from the more conserved motifs found first. Locations of the previously located motifs can be assumed as prior information, similar to our information of a relevance of each sample.

The way in which MEME erases a motif is the following. Assume a vector W of length n, where n is the number of the samples in the input dataset. The value W_i represents prior information about suitability of i-th sample as a motif occurrence. The values may range over interval $\langle 0, 1\rangle$. The higher the value, the more appropriate sample for motif occurrence.

After a motif is discovered, vector of values A is estimated, where A_i gives the probability that i-th sample is not part of a motif occurrence. The previous value of W_i is then updated by multiplying it by A_i. These values are used in reestimating the nucleotide frequencies in equation 2.25. Instead of summing the offset probabilities Z_{ij}, the weighted offset probabilities W_iZ_{ij} are summed.

To understand how the weighting scheme erases previously discovered motifs, suppose that MEME has discovered one motif and is looking for the second. Samples, that are more probable to be parts of a motif occurrence have smaller W value and therefore do not contribute to estimation of the motif parameters much. Considerably larger contribution is from samples, that are not parts of a motif occurrence and therefore have higher W values.

Prior information about GC clusters location can be used in similar fashion. We have initialised W values with relevance, so that initial W_i value responds to the relevance of the i-th sample. Samples from AT rich areas should have lower W value and therefore contribute less than samples from GC clusters with W values close to 1.

4.3.3 Selection of the initial samples

We have estimated all model parameters, except motif component θ_1 and quantity of motif occurrences λ_1. Here we propose method of selecting them systematically.

We are discovering motifs in GC islands, so the initial motif component should respond to composition of the sequences in these areas. We have taken all samples from the input sequences that have relevance at least 0.5. Each sample have been used to prepare an initial motif model. A sample $x = x_1, x_2 \dots x_w$ is converted to the motif emission probability table $\theta_1 = (f_1, f_2 \dots f_w)$ column-by-column. For constant m we set them as

$$f_{ij} = \begin{cases} m & x_i = j \\ \frac{1-m}{L-1} & x_i \neq j \end{cases}$$

The value m has been set to 0.6 in our analysis.

Another unknown parameter is the initial quantity of motif occurrences λ_1. We examined several values for each motif model. The lowest considered value was $\frac{\sqrt{M}}{m}$ where M is the number of the GC clusters in the input sequences and m is the number of examined samples that have relevance higher than 0.5. The highest value has been set to $\frac{M}{m}$. The bounds express a belief, that the discovered motif should be presented in at least $\sqrt{M}$ GC clusters, but there should be no more then one occurrence of the motif per GC island. The sampling of the quantity was done in a geometrically increasing series, based on the observation (24), that initial parameters, where λ_1 was within a factor of 2 of the correct value were sufficient for EM to converge.

Each combination of a generated motif model and the frequency of motif occurrences has been joined with previously estimated parameters θ_2, θ_3, λ_2 and λ_3 to prepare a new model. Each model has been improved by one EM iteration and a log-likehood value of the improved model has been calculated. Models with highest log-likehood value are considered as the best candidates for further analysis.

We have selected the ten best candidates as initial parameters for the EM algorithm. The E-step and the M-step of EM has been executed repeatedly, until the change in θ_1 under the Euclidean distance falls bellow 10^{-6}.

4.4 Experiment

4.4.1 Synthetic data

We have prepared new datasets, each consisting of 100 sequences. At first, the sequences of length 164 (average length of the AT regions) were generated from the distribution of the AT rich state (θ_2). Each sequence has been divided at random position into two parts, a_1 and a_2. A new GC cluster c of length 39 (average length of the GC clusters) has been generated from the distribution of the GC rich state (θ_3). The sequences have been combined into a single sequence a_1ca_2. The nucleotides in the GC clusters have been generated independently, thus we can not hope to find significant motifs in them. Therefore we modified the GC clusters by changing a random substring to a motif occurrence.

At first we have prepared a motif model of length w. We have used $w = 10$ in our experiments. For position i of the motif we have generated one nucleotide r_i from the distribution θ_3. The frequency of j-th nucleotide type on the i-th position was initialised as

$$f_{ij} = \begin{cases} m & \text{if } j = r_i \\ \frac{1-m}{L-1} & \text{if } j \neq r_i \end{cases}$$

We have used value $m = 0.9$ in our experiments.

A motif instance $x = (x_1, x_2, \ldots, x_w)$ was generated nucleotide by nucleotide. The nucleotide x_i has been chosen randomly from distribution f_i. After generation, we selected a random position in a GC cluster and changed the substring starting at the position to the motif occurrence x.

We tried different number of motif instances in the dataset. The lowest quantity has been determined, so that every tenth GC cluster contains motif occurrence. The highest quantity refers to that motif instance has been located in each GC cluster. For each quantity we have generated 100 datasets, each contained different motif. We executed the reimplemented MEME algorithm on these datasets as well as our modified algorithm, the GC MEME. The result was considered successful, if the dominant nucleotide at each position i of the found motif corresponds to the nucleotide with the highest frequency in

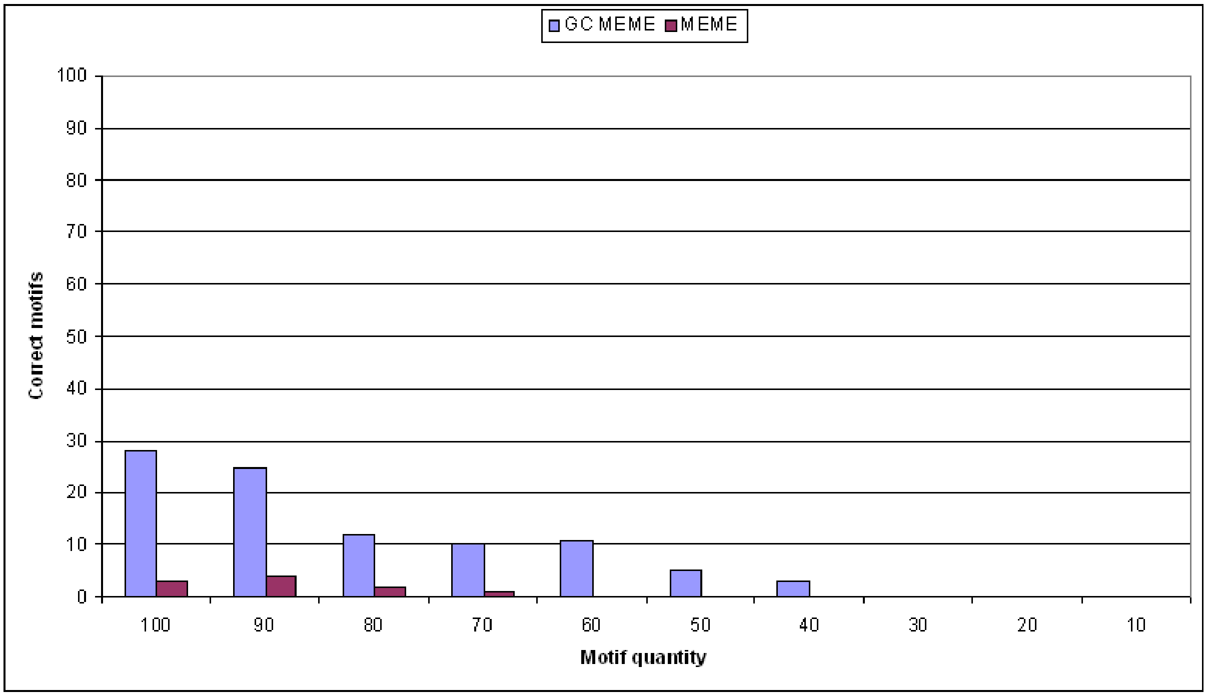

Figure 4.2: Comparison of the original and the adjusted MEME algorithm

distribution f_i. The results are shown in the histogram 4.2.

The results show that the GC MEME was significantly better. In fact, the MEME algorithm was able to correctly identify motif only in the 0.1% datasets. To identify the problem that lead to this high error rate, we explored discovered motifs. They were vastly enriched by cytosine and guanine. An example of such a motif is in the figure 4.3.

The background of the MEME model is composed of one state that is not able to differentiate between AT and GC rich areas. Due to a majority of the AT rich areas, the

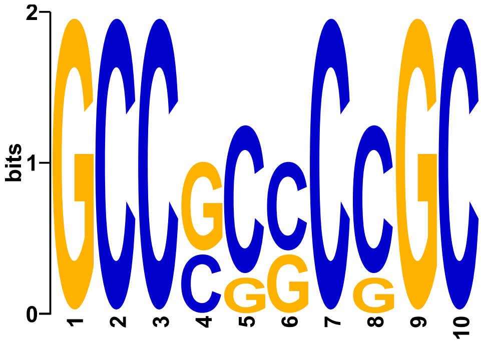

Figure 4.3: An example of a motif found by the MEME algorithm

background distribution is estimated so that adenine and thymine have higher background frequency. Samples, that contains these nucleotides are therefore disadvantaged against samples composed purely from cytosine and guanine, that occur relatively frequently in the GC rich areas.

The modified algorithm with two state background model achieved much better results. However it was still able to correctly identify only 28% samples at the highest motif quantity.

4.4.2 Real data

We have executed the adjusted MEME algorithm on the real annotated sequences from mitochondrial genomes. The results were surprising, all candidates have converged to a similar model. Each position of the motif model has similar distribution with dominant G nucleotide. Closer look at the GC clusters reveals the problem.

The GC clusters contain many stretches that are composed entirely of cytosine or guanine. Assume a sample T, composed of six guanine nucleotides, "GGGGGG", and a sample S, with varied structure, for example "CGAGCA". Both samples should contribute to estimation of the motif model equally. Unfortunately, samples from the input dataset are not independent and so the repetitive sequences have an advantage compared to the varied.

Assume, that cluster T occurs at i-th and S on the j-th position of the input sequence The input sequence is broken up into overlapping samples of length w. Here we present several selected samples, where the "." sign constitutes for arbitrary nucleotide.

```
...                                 ...
sample i-5  .....GGGGGG             sample j-5  .....CGAGCA
sample i-4  ....GGGGGG.             sample j-4  ....CGAGCA.
sample i-3  ...GGGGGG..             sample j-3  ...CGAGCA..
sample i-2  ..GGGGGG...             sample j-2  ..CGAGCA...
sample i-1  .GGGGGG....             sample j-1  .CGAGCA....
sample 1    GGGGGG.....             sample j    CGAGCA.....
....                                ....
```

The samples of T cluster are clearly more similar to each other as samples of S cluster and therefore seem more conserved. This reflects on the estimation of the likehood function that would prioritise repetitive cluster against varied.

The problem should be eliminated by additional modification in the algorithm. One of the possibilities is to reduce the relevance of the samples, that contain repetitive nucleotides. An another possibility is a refinement of the calculation of a sample background probability. We revise our assumption about background nucleotides so that nucleotides are not generated independently, but some combinations of consecutive nucleotides are more probable than others. In this way, the consecutive pairs of nucleotides in the repetitive sequences would have greater probability of arising from background model and therefore they should contribute less to the estimation of the motif model.

Conclusion

In this thesis we proposed several modifications of the motif discovery MEME algorithm. The changes were made in order to meet two objectives. First we investigated ways for speeding up the algorithm using a new method of selection of initial parameters. Our approach helps to decrease the number of required iterations of the EM algorithm, which is the core algorithm of MEME. Our second contribution adapts MEME for the specific structure of the mitochondrial genomes of yeasts, which was achieved by changes in the MEME probabilistic model.

The new method of selection of initial parameters is based on the concept of probes. In our framework, we probe is as a combinations of two nucleotides at a certain distance. The probes that occur in the input sequences more frequently than expected are good candidates for further examination based on the assumption that their increased frequency is caused by contribution from motif instances.

We have evaluated our approach use a dataset of known transcription factor binding site motifs embeded in randomly generated background sequence. We have varied the number of motif occurrences and dataset size. We propose that evaluating 10 best probes, each within the length 10, should be sufficient for capture the motif presented in the dataset.

The motif frequency is unknown parameter of the model and therefore must be estimated before EM iterations. The MEME algorithm uses multiple enumerated values. We based our estimation of the value of the difference between the observed and expected occurrences of the probe. The experiments indicate that using this estimate, the number of enumerated values can be reduced to three.

Perhaps the most important part of the initialisation of the model parameters is estimation of the appropriate initial motif model. We propose a method that examines the neighbourhood of the probe locations to establish the distributions of the motif positions. Incorporation of the estimated motif quantity significantly improves this method as confirmed by experiments.

Overall our proposed changes in initialization decreases the running time required in the initialization phase of the EM algorithm and even decrease the number of iterations of the EM required for convergence.

Another changes has been made in the MEME model. Experiments show that the background component originally composed of one state, is not able to take into account statistical differences between GC clusters and surrounding sequence in yeast mitochondrial genomes. Therefore we extended it by addition of a new state which is responsible for generation sequences similar to GC clusters. Since we were mainly interested in the motifs in the regions significantly enriched by cytosine and guanine (GC clusters), we incorporate prior information in order to direct motif discovery to the GC clusters.

The modified algorithm achieved significantly better results on the synthetic data than the reimplemented MEME algorithm. However experiments shows that it does not provide sufficient results on the real data. The algorithm has great tendency to find repetitive sequences such as nucleotide stretches that are composed entirely of one nucleotide type. Repetitive sequences are quite common in the GC clusters and that made our results defective.

The problem with repetitive sequences that we have found should be taken into account in the future work. We propose two changes in the algorithm to address this problem. One of them is decreasing of the relevancy of repetitive sequences. Another suggestion is to establish more accurate calculation of the background probability.

Our work on initialisation also suggests interesting avenues for future research. In the thesis we have examined probes as pairs of nucleotides. The probe concept can be easily expanded to a combination of more nucleotides that should lead to better accuracy of the algorithm. We can also compare nucleotides around placement of the probe and search for locations that contain some nucleotide with significantly higher frequency than expected. In this way we can estimate motif width as well.

Resume

V tejto práci sme skúmali MEME algoritmus, ktorý je určený na vyhľadávanie motívov v biologických sekvenciách. Algoritmus sme implementovali so zmenami, ktoré viedli k naplneniu dvoch základných cieľov, a to zrýchlenie algoritmu a jeho špecializovanie na vyhľadávanie motívov v mitochondriálnej DNA, ktorá sa vyznačuje netradičnou štruktúrou.

Jadrom MEME je takzvaný expectation-maximization (EM) algoritmus. Proces začína odhadnutím počiatočného modelu, ktorý popisuje vstupné sekvencie. Parametre modelu sú následne iteratívne vylepšované tak, aby každý nový model zvyšoval vierohodnosť vygenerovania týchto sekvencií. Algoritmus zaručuje konvergenciu do lokálneho maxima, zvolenie vhodných počiatočných parametrov je preto kľúčovou úlohou pre nájdenie globálneho maxima, a teda aj najlepšieho motívu.

MEME odhaduje počiatočný model z náhodne vybraných podreťazecov zo vstupných sekvencií. V práci navrhujeme deterministický výber počiatočného modelu založenom na koncepte takzvaných sond.

Sondou budeme nazývať dvojicu nukleotidov, ktoré sú od seba vzdialené v presne určenej vzdialenosti. Sondy, ktoré majú v sekvenciách viac výskytov, ako by sa očakávalo sú dobrými kandidátmi na preskúmanie. Vychádzame z predpokladu, že zvýšený počet výskytov je zapríčinený dodatočnými výskytmi motívu, v ktorom sa daná sonda nachádza.

Pre naše experimenty sme použili motívy z databázy Jaspar, ktoré opisujú modely pre publikované miesta viazania transkripčných faktorov. Tieto motívy sme pridávali v rôznych množstvách do vstupných sekvencií rôznych dĺžok a skúmali sme, ako presná bude naša heuristika. Výsledky ukázali, že preskúmanie prvých 10 sond s maximálnou dĺžkou 10 by malo byť dostatočné, aby sme zachytili sondu, ktorá patrí motívu.

Jedným z parametrov počiatočného modelu, ktorý ovplyvňuje kvalitu výsledku hľadania, je počet výskytov motívu v sekvenciách. MEME problém určenia počiatočnej hodnoty rieši tak, že skúša viacero rôznych hodnôt. Ak však preskúmame počet výskytov sondy v sekvenciách a porovnáme ho s tým, koľko hodnôt očakávame, môžeme odhadnúť, koľko výskytov patrilo inštancii motívu. Pomocou tohto odhadu je možné zmenšiť počet počiatočných hodnôt parametra na tri.

Najdôležitejšou časťou inicializácie je odhadnutie počiatočného modelu motívu. V práci navrhujeme metódu, ktorá skúma okolia výskytov sond, a takýmto spôsobom je možné odhadnúť frekvencie nukleotidov na každej z pozícií motívu. Počiatočný model motívu sa dá ešte vylepšiť tým, že do výpočtu zahrnieme aj odhadnutý počet výskytov motívu.

Experimenty potvrdili, že týmito zmenami v inicializácii sa zníži nielen čas behu inicializačnej fázy algoritmu, ale aj počet iterácií potrebných pre EM konvergenciu.

Skúmaním sekvenovaných mitochondriálnych genómov kvasiniek sa odhalila ich zaujímavá štruktúra. Väčšina genómu je bohatá na nukleotidy adenín a tymín, niektoré časti sú ale výrazne obohatené o nukleotidy cytozín a tymín, nazývané aj GC ostrovy. Experimenty ukázali, že pôvodný model nebol schopný zachytiť štatistické rozdiely medzi týmito oblasťami. V práci sme preto navrhli zmeny v modely MEME algoritmu, ktoré ho špecializujú na úlohu vyhľadávania motívov v mitochondriálnych genómoch.

V práci nás najviac zaujímali motívy nachádzajúce sa v GC ostrovoch. Do algoritmu sme teda vložili dodatočnú informáciu o ich umiestneniach. Tá bola využitá na nasmerovanie procesu vyhľadávania motívov do GC ostrovov.

Upravený algoritmus si na vygenerovaných sekvenciách počínal podstatne lepšie, ako originálny MEME. Výsledky už ale boli horšie, keď sme ho odskúšali na reálnych dátach. Algoritmus mal tendenciu nájsť rovnomerné klastre, teda úseky tvorené iba jedným nukleotidom. Tieto úseky sú v GC ostrovoch pomerne časté, a to výrazne zhoršilo výsledky.

V práci sme navrhli dve možné riešenia problému. Jednou z nich je zníženie relevancie rovnomerných klastrov. Ďalšou možnosťou je presnejšie určovanie pravdepodobnosti, že oblasť v modely nie je výskytom motívu.

Naša práca umožňuje zaujímavé miesta na rozšírenie. V prípade sond sme napríklad uvažovali iba dvojice nukleotidov. Rozšírenie o viacero nukleotidov by mohlo viesť ku zvýšeniu presnosti tohto prístupu. Ďalšou možnosťou je podrobnejšie pozorovanie okolia výskytov sondy, ktoré by mohlo viesť ku odhadu šírky motívu.

Bibliography

[1] Audrey Gasch and Michael Eisen. Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome Biology*, 3:1–22, 2002. 10.1186/gb-2002-3-11-research0059.

[2] Albin Sandelin, Wynand Alkema, Pär Engström, Wyeth W. Wasserman, and Boris Lenhard. Jaspar: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Research*, 32(suppl 1):D91–D94, 2004.

[3] Leonid A. Mirny and Mikhail S. Gelfand. Structural analysis of conserved base pairs in protein DNA complexes. *Nucleic Acids Research*, 30(7):1704–1711, 2002.

[4] Alan Moses, Derek Chiang, Manolis Kellis, Eric Lander, and Michael Eisen. Position specific variation in the rate of evolution in transcription factor binding sites. *BMC Evolutionary Biology*, 3(1):19, 2003.

[5] Amos Bairoch. Prosite: a dictionary of sites and patterns in proteins. *Nucleic Acids Research*, pages 2241–2245, 1991.

[6] Gunnar von Heijne. A new method for predicting signal sequence cleavage sites. *Nucleic Acids Research*, pages 4683–4690, 1986.

[7] Thomas D. Schneider and R.Michael Stephens. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Research*, 18(20):6097–6100, 1990.

[8] Martha L. Bulyk, Philip L. F. Johnson, and George M. Church. Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors. *Nucleic Acids Research*, 30(5):1255–1261, 2002.

[9] Tsz-Kwong Man and Gary D. Stormo. Non-independence of mnt repressor–operator interaction determined by a new quantitative multiple fluorescence relative affinity (qumfra) assay. *Nucleic Acids Research*, 29(12):2471–2478, 2001.

[10] Qing Zhou and Jun S. Liu. Modeling within-motif dependence for transcription factor binding site predictions. *Bioinformatics*, 20(6):909–916, 2004.

[11] Martin C. Frith, Neil F. W. Saunders, Bostjan Kobe, and Timothy L. Bailey. Discovering sequence motifs with arbitrary insertions and deletions. *PLoS Comput Biol*, 4(5):e1000071, 05 2008.

[12] Gary A. Churchill. Stochastic models for heterogeneous DNA sequences. *Bulletin of Mathematical Biology*, 51(1):79 – 94, 1989.

[13] Kenzie D MacIsaac and Ernest Fraenkel. Practical strategies for discovering regulatory DNA sequence motifs. *PLoS Comput Biol*, 2(4):e36, 04 2006.

[14] H O Smith, T M Annau, and S Chandrasegaran. Finding sequence motifs in groups of functionally related proteins. *Proceedings of the National Academy of Sciences*, 87(2):826–830, 1990.

[15] Marie-France Sagot, Alain Viari, and Henri Soldano. Multiple sequence comparison: A peptide matching approach. In Zvi Galil and Esko Ukkonen, editors, *Combinatorial Pattern Matching*, volume 937 of *Lecture Notes in Computer Science*, pages 366–385. Springer Berlin / Heidelberg, 1995.

[16] Inge Jonassen. Efficient discovery of conserved patterns using a pattern graph. *Computer applications in the biosciences : CABIOS*, 13(5):509–522, 1997.

[17] Modan Das and Ho-Kwok Dai. A survey of DNA motif finding algorithms. *BMC Bioinformatics*, 8(Suppl 7):S21, 2007.

[18] CE Lawrence, SF Altschul, MS Boguski, JS Liu, AF Neuwald, and JC Wootton. Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment. *Science*, 262(5131):208–214, 1993.

[19] Martin C. Frith, Ulla Hansen, John L. Spouge, and Zhiping Weng. Finding functional sequence elements by multiple local alignment. *Nucleic Acids Research*, 32(1):189–200, 2004.

[20] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.

[21] Kazuhito Shida. Gibbsst: a gibbs sampling method for motif discovery with enhanced resistance to local optima. *BMC Bioinformatics*, 7(1):486, 2006.

[22] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis*, chapter 3.1. Cambridge Universiy Press, 1. edition, 1998.

[23] Timothy L. Bailey and Charles Elkan. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning*, 21:51–80, 1995. 10.1023/A:1022617714621.

[24] Timothy Bailey and Charles Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, pages 28–36. AAAI Press, 1994.

[25] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):pp. 1–38, 1977.

[26] Kullback S. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[27] Miklos de Zamaroczy and Giorgio Bernardi. The GC clusters of the mitochondrial genome of yeast and their evolutionary origin. *Gene*, 41(1):1 – 22, 1986.

[28] Christiane Bouchier, Laurence Ma, Sophie Creno, Bernard Dujon, and Cecile Fairhead. Complete mitochondrial genome sequences of three nakaseomyces species reveal invasion by palindromic gc clusters and considerable size expansion. *FEMS Yeast Research*, 9(8):1283–1292, 2009.

[29] Ariel Prunell, Helena Kopecka, François Strauss, and Giorgio Bernardi. The mitochondrial genome of wild-type yeast cells: V. genome evolution. *Journal of Molecular Biology*, 110(1):17 – 47, 1977.

[30] Giorgio Bernardi. Lessons from a small, dispensable genome: The mitochondrial genome of yeast. *Gene*, 354:189 – 200, 2005. Cross-Talk between Nucleus and Organelles.

[31] Juraj Mestanek and Peter Peresini. Evolutionary history of GC clusters. Technical report, Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava, 2009.

[32] Howard S. Bilofsky and Burks Christian. The GenBank: genetic sequence data bank. *Nucleic Acids Research*, 16(5):1861–1863, 1988.

[33] Miklos Csuros. Maximum-scoring segment sets. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1:139–150, 2004.

[34] Stuart Jonathan Russell and Peter Norvig. *Artificial Intelligence : A Modern Approach*, chapter 15.2. Upper Saddle River, EUA : Prentice-Hall, 2003.