Chapter 2

Metamotifs - a generative model for building families of nucleotide position weight matrices

2.1 Background

[1] A fundamental difficulty in studying DNA specificity of TFs is the absence of a simple, universal recognition code from the protein sequence or tertiary structure of the TF to its DNA recognition motif (Smith, 1998). Comparative studies of TF domains and their crystal structures with bound DNA have shown certain recurring rules for protein-DNA interactions (Jones et al., 1999; Kono and Sarai, 1999; Nadassy et al., 1999), for instance commonly occurring hydrogen bond mediated interactions between the base guanine, and arginine, lysine, histidine or serine residuess (Luscombe and Thornton, 2002). However, the stronger patterns predictive of DNA specificity of proteins are highly TF domain family specific (Kono and Sarai, 1999; Luscombe and Thornton, 2002). That these interactions are domain specific, and sometimes non-additive (Badis et al., 2009; Benos et al.,

[1] This chapter, and the two following two, were partly published in BMC Bioinformatics (Piipari et al., 2010a), by the author of this PhD thesis (MP), Dr. Thomas Down (TD), and my thesis supervisor Dr. Tim Hubbard (TH). Authors' contributions are as follows: TH, TD and MP conceived the work, MP developed the software, performed the tests and wrote the manuscript.

2002a), should not come as a surprise; protein and DNA interactions form a dynamic three dimensional network of contacts between the protein residues, the DNA sugar–phosphate backbone, bases and water residues in the binding interface (Luscombe and Thornton, 2002). Substantial conformational changes of both the protein and the DNA also often occur upon binding (Kim, 1995; Percipalle et al., 1995).

Even though a universal recognition code of protein DNA binding is unlikely to surface, familial patterns of DNA binding specificity can still be made use of to provide biological insight about newly presented data. The interaction rules of the DNA-binding residues are understood well in the case of some extensively studied domains like Cys_2His_2 zinc fingers (Wolfe et al., 2000). The DNA specificity of a Cys_2His_2 domain can be predicted based on sequence (Benos et al., 2002b; Kaplan et al., 2005; Mandel-Gutfreund et al., 2001; Persikov et al., 2008), and altogether new transcription factors can be engineered by mutating the DNA binding residues (Pabo et al., 2001). More interestingly from the point of view of my work, however, familial patterns of DNA specificity can be taken to infer TFBS motifs from genomic sequence with greater sensitivity. Several algorithms have been designed that take into account previous knowledge of TF domain DNA specificity to find motifs which fit familial patterns, or to label newly discovered motifs to TF families with classification methods (Narlikar et al., 2006; Sandelin and Wasserman, 2004; Xing and Karp, 2004).

2.1.1 Previous work on motif family models

The most widely applicable model for short regulatory motifs is the position weight matrix, or PWM (see Section 1.2.1), originally introduced by Stormo et al. (1982). Methods have been developed for comparing and clustering PWMs. The earliest such methods were made for protein domain model comparison (Pietrokovski, 1996). In the case of DNA motifs, clustering can be used to infer information about possible function of *de novo* predicted motifs, such as to find clusters of closely related motifs to known data. Although DNA binding domains vary widely, familial tendencies exist in DNA sequence motifs that are predictive of the family of transcription factors which bind them (Narlikar et al.,

2006; Narlikar and Hartemink, 2006). This makes clustering useful for inferring potential binding partners for discovered motifs of interest.

Familial binding profiles (FBP) offer perhaps the earliest solution for summarising familial patterns in nucleotide PWMs (Sandelin and Wasserman, 2004). FBPs are weighted averages of aligned sets of motifs. All motif pairs in the set are aligned with a variant of the Needleman & Wunsch global alignment algorithm (Needleman and Wunsch, 1970), using the score defined in Equation 2.1 to minimise the sum of squared deviations between the aligned motif columns amongst a familial alignment of PWMs, allowing for a single gap (with a stringent but arbitrarily chosen gap opening penalty). The significance of scores is measured with an empirical distribution of motif pair scores derived from shuffled motifs of the same length (Sandelin and Wasserman, 2004). Motifs are then added to a multiple alignment in the order of decreasing significance, and finally the motif columns are averaged, with contribution of each motif V weighted according to $w_V = 1 - p_v$, where p_v is the average of p-values of motif V with all the other motifs.

$$S = 2 - \sum_{b \in \{A,C,G,T\}} (M_b - N_b)^2 \tag{2.1}$$

FBPs for 11 metazoan transcription factor families are made available through the JASPAR motif database (Portales-Casamar et al., 2010). However, the FBP-based approach suffers from certain inherent limitations; Firstly it is not a probabilistic method but uses an arbitrary distance metric between motif columns, necessitating an empirical significance score computation and an arbitrary weighting of motif contributions to the FBP. Secondly, a global alignment is assumed between all motif columns, which means that only patterns common to all members of the family can be reliably modeled in this fashion. Sandelin and Wasserman (2004) only present FBPs for a small number of metazoan specific groups of DNA binding domains (11 FBPs, built from a total of 63 closely related motifs). Incidentally, many of the DNA binding domains in these 11 (e.g. ETS, Rel, MADS) have been classified as 'highly specific' to their DNA binding sites already by Luscombe et al. (2001), meaning that TFs in these families have a closely similar distribution of binding site specificities (motifs) with little variation.

More generally, motif comparison methods also suffer from the absence of a natural distance metric between motifs, although many different metrics have been proposed for this problem. For instance, a χ^2-based distance metric was found an effective measure by Kielbasa et al. (2005). A metric based on Pearson correlations of motif columns was also described in the same publication. Various other distance metrics were suggested and systematically evaluated in a study by Mahony et al. (2007), where a sum of squared deviations based metric was found to be the best single metric. The asymptotic covariance between hits of two motifs in an infinitely long sequence parameterised by its nucleotide content has also been applied as a distance measure (Pape et al., 2008). The most recent motif distance metric and clustering methods are probabilistic and draw special attention to the uncertainty in motif comparison and the importance of high-information columns in measuring distances of sequence motifs: a Bayesian probability distance metric between motif columns (Habib et al., 2008) and a fuzzy integral based metric (Garcia et al., 2009). In this work I also explore a probabilistic solution for comparing motifs. Unlike any of the above motif-to-motif distance work, I however do not apply the developed method to a motif clustering problem. Instead, I attempt to solve the supervised learning problem of classifying motifs to their TF domain families probabilistically (Chapter 4). Classification based learning can be arguably more informative when predicting the likely function of motifs. This is because assigning a motif to a motif family has an associated uncertainty. Therefore finding closely similar known motifs by clustering does not always allow precise conclusions to be made regarding the binding partner of a discovered motif.

Supervised learning strategies have been applied to classify motifs and infer motifs similar to previously known motifs from novel sequences. Self-organising maps (Kohonen and Somervuo, 2002) have been applied for classification of binding sites for the purposes of semi-supervised motif inference in the SOMBRERO algorithm (Mahony et al., 2005a). Other notable methods include a Sparse Multinomial Regression (SMLR) based binding site sequence classification described in Narlikar and Hartemink (2006), and an application of this method to motif inference; The motif inference program PRIORITY assigns an SMLR-derived prior probability for each sequence position for its potential to fit a motif of a given

transcription factor family (Narlikar et al., 2006).

I present here a probabilistic model for describing motif families and measuring relatedness of sequence motifs – the metamotif. Metamotifs can be used to summarise gapless alignments of motifs of a given length, similar to an FBP. In contrast to the FBP framework introduced by Sandelin and Wasserman (2004), I do not model the recurring patterns found amongst a related set of motifs necessarily as a single motif alignment. Furthermore, the metamotif includes a vector of column wise mean nucleotide weights, as well as a variance parameter for each column. Variance is not modelled for example by the FBP or other non-probabilistic methods. Inclusion of motif column variances as part of the model makes it unnecessary to derive empirical significance estimates of motif similarity. In this respect a metamotif is similar to the hierarchical profile hidden Markov–Dirichlet multinomial model used by MotifPrototyper (Xing and Karp, 2004): both describe familiar prototypes of PWMs that are estimated probabilistically with a sequence of position specific probability distributions and can yield a Bayesian prior on the weight matrix columns (a 'structural prior' for the weight matrices in the terminology used by Xing and Karp (2004)). In contrast to MotifPrototyper, however, the metamotif inference algorithm I developed (Section 2.2.4) can account for intra-motif structure such as repeating or palindromic segments by treating motifs as a series of potentially several metamotif instances (i.e. learning several prototype patterns rather than only one), and positions emitted by a background model. In other words, in our framework, not all positions are generated from a single metamotif, and I additionally model some motif positions as noise emitted by a background model.

2.2 The metamotif

A metamotif is a generative model for PWM motif columns that can be used to represent a gapless alignment of position weight matrices. For each PWM position i (multinomial column) there exists a Dirichlet distribution in the metamotif column at position i. A metamotif is therefore a parameter configuration for a product Dirichlet distribution where position i of the motif alignment model corresponds to parameters α_i. More intuitively, consider that in a metamotif,

nucleotides at all positions have an associated average weight (depicted in Figure 2.1A as the symbol heights) and a variance (the error bars). It is in other words a probability distribution over PWM motifs of a given length. A metamotif of length k therefore allows drawing motifs of length k from it (Figure 2.1B), and querying for the probability of the metamotif being the source distribution for any motif of the same length. This is analogous to computing a probability score for a sequence k-mer to measure the probability of the k-mer having been generated by a PWM.

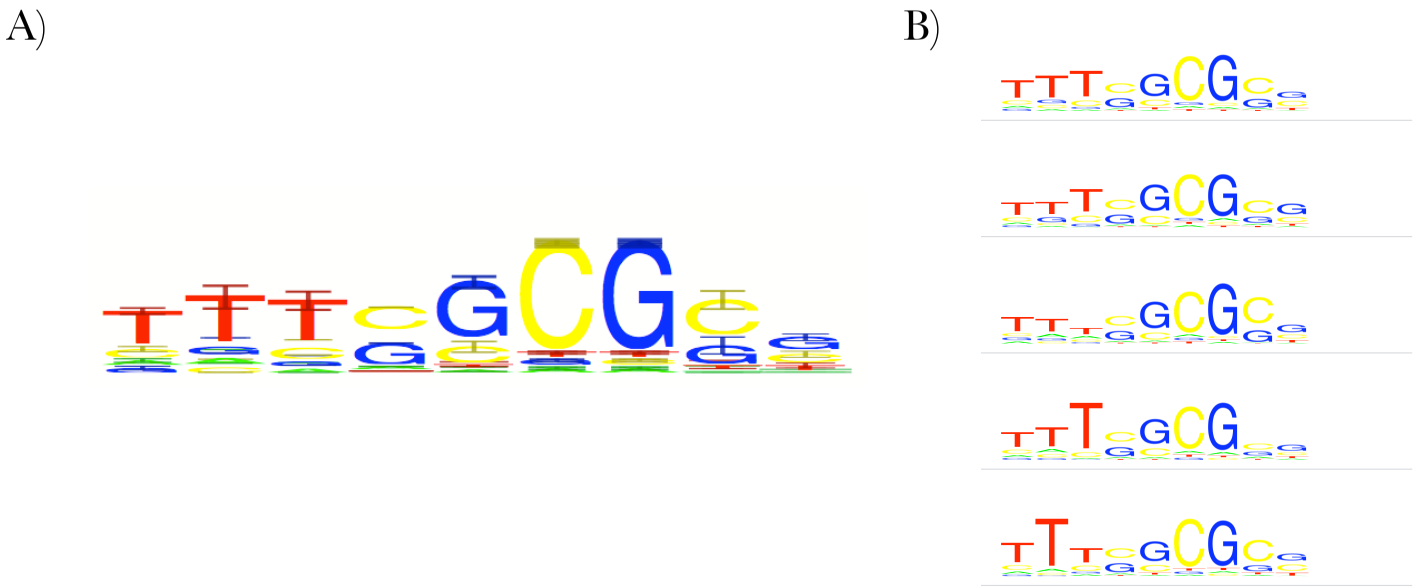


Figure 2.1: A forkhead-like metamotif (inferred from an alignment of motifs) is shown alongside selection of motif samples drawn from it. The wider error bars (representing 95% confidence intervals of nucleotide weights) of the thymine-rich $5'$ end of the metamotif is found consistent with the variation in the motif column heights. A) metamotif is a column-wise model with average nucleotide weights and variance associated per nucleotide column. B) A metamotif is a probability distribution from which motifs of the same length can be drawn.

Below I first formally define the metamotif (Section 2.2.1) and present a simple maximum likelihood method for estimating metamotifs from aligned motif data. In Section 2.2.2 I present a form of visualisation for the metamotif akin the sequence logo (Schneider and Stephens, 1990), and then expand the use of the model beyond simply constructing metamotifs from aligned motifs (Section 2.2.4). This expansion is made possible by a Monte Carlo metamotif inference algorithm that simultaneously estimates multiple weakly represented metamotifs from a potentially large set of motifs.

2.2.1 Formulation of the model

A metamotif α is a matrix of L columns, each defining a Dirichlet distribution over $\mathbb{R}^K$ where K is the size of the alphabet (Equation 2.2).

$$\alpha = \begin{pmatrix} \alpha_{11} & \dots & \alpha_{1L} \\ \vdots & & \vdots \\ \alpha_{K1} & \dots & \alpha_{KL} \end{pmatrix} \tag{2.2}$$

A motif $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ is a set of column vectors over the same alphabet. The probability of observing the column $\mathbf{x}_i$ from the metamotif α is given by the density of the Dirichlet distribution with parameters α_i at weights $\mathbf{x}_i$ (Equation 2.3). The normalising constant $B(\alpha)$ is the multinomial beta function, expressed in Equation 2.4 via the Gamma function.

$$\mathbb{P}(\mathbf{x}_i|\alpha_i) = \mathrm{Dir}(\mathbf{x}_i;\alpha_i) = \frac{1}{B(\alpha)} \prod_{j=1}^{K} x_{ij}^{\alpha_{ij}-1} \tag{2.3}$$

$$B(\alpha) = \frac{\prod_{j=1}^{K} \Gamma(\alpha_j)}{\Gamma(\sum_{j=1}^{K} \alpha_j)} \tag{2.4}$$

The log probability of observing a motif of length L is then given by Equation 2.5.

$$\log \mathbb{P}(\mathbf{X}|\alpha) = \sum_{i=1}^{L} \log(\mathrm{Dir}(\mathbf{x}_i;\alpha_i)) \tag{2.5}$$

To motivate the use of the metamotif we note that the metamotif column α_i can be understood as a combination of the mean nucleotide weights $\mathbb{E}[x_{mk}]$ and precision $\alpha_{0m} = \sum_{j=1}^{K} \alpha_j$ (Equation 2.6) where $m \in [1, M]$ and $k \in [1, K]$.

$$\mathbb{E}[x_{ij}] = \alpha_{ij}/\alpha_{0j} \tag{2.6}$$

2.2.2 Visual representation of the model

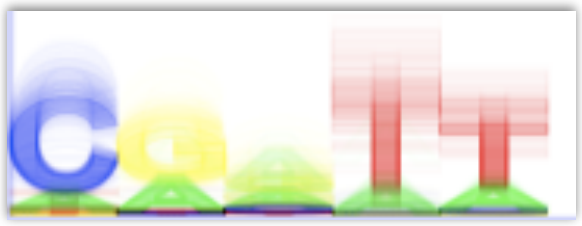
The visual representations I developed for the metamotif model are both based on the sequence logo (Schneider and Stephens, 1990). The metamotif visualisation was implemented as part of the iMotifs sequence motif visualisation environment (Piipari et al., 2010b) with Apple's C-based Quartz API and the Objective-C based Cocoa drawing APIs. Metamotif model visualisation was in fact originally implemented in a Java based cross-platform motif visualisation tool mXplor, which I created as a precursor to iMotifs (available openly at `http://www.github.com/mz2/mxplor`). The representation evolved from a 'fuzzy sequence logo', where a number of sequence logos are overlaid on top of each other (Figure 2.2A), to a sequence logo with confidence intervals being drawn on the motifs (Figure 2.2B). Notably iMotifs supports both the error bar and fuzzy motif representations.

Both visual forms shown in Figure 2.2 communicate the mean weights $\mathbb{E}[\mathbf{X}|\alpha]$ and precision α_0 aspects of the metamotif. A sequence logo is drawn for PWM with nucleotide weights $\mathbb{E}[\mathbf{X}|\alpha]$. In the error bar enabled sequence logo in Figure 2.2B the error bars are shown to highlight 95% confidence intervals of nucleotide weights of the Dirichlet density at α_i for each symbol (Figure 2.2B).

2.2.3 Aligning motifs and estimating metamotifs from a motif multiple alignments

Given that a metamotif is a probability distribution over motifs of length k, it should be possible to estimate a metamotif from a series of aligned motif columns of matching length (see for example Figure 2.1B). Indeed, during my project I firstly designed a simple maximum likelihood metamotif inference algorithm for

A)



B)

C)

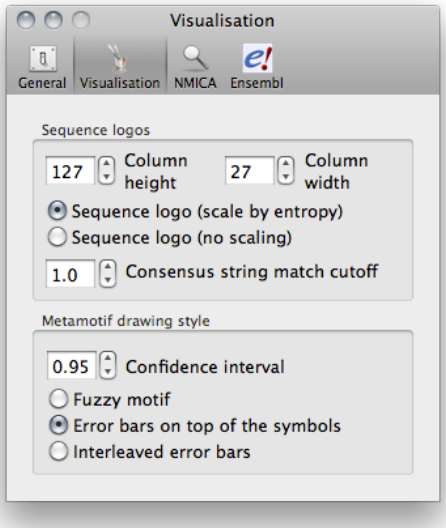


Figure 2.2: Visual representations of metamotifs. A) A 'fuzzy sequence logo' representation of a metamotif drawn with mXplor. One hundred samples are drawn per column, and sequence logos of the resulting PWMs are overlaid with low opacity on top of each other. The symbols in the sampled PWMs are ordered according to the decreasing nucleotide weight of the average weights in the distribution. B) Metamotif represented by a sequence logo with error bars (5% – 95% confidence intervals are presented with the error bars). C) The confidence intervals presented for a metamotif, i.e. the 'height' of the error bars in (B), can be configured in iMotifs.

the purpose. It is described in brief in Figure 2.3.

Firstly, a distance distribution is computed between the input motifs according to the column-wise sum of squared differences (SSD) motif distance metric from Down et al. (2007), which is noted below in Equation 2.7. P and Q are distributions from the two compared motifs, and ϵ is an adjustable modifier on the exponent. When it has the value 1.0, the distance computed is the Cartesian distance. Similar to Down et al. (2007) I use $\epsilon = 2.5$.

$$D(P||Q) = (\sum_{s \in A}(P(s) - Q(s))^2)^{\epsilon/2} \tag{2.7}$$

When comparing the distance, all possible offsets with at least one overlapping column are considered between motif pairs (the unmatched columns are treated as a multinomial distribution with uniform nucleotide weights $[0.25, 0.25, 0.25, 0.25]$). Then, beginning from the closest motif pair, motifs are progressively added to the alignment, one by one in the order of increasing distance to motifs already present in the alignment. This is analogous to the progressive multiple alignment strategy used in many protein sequence multiple alignment algorithms (Chenna et al., 2003; Notredame et al., 2000). The resulting gapless alignment is simply defined by the offsets and reverse complement operations required to minimise the distance between the closest pairs (reverse-complementing motifs, i.e. allowing matches on either strand, is optional). Computing the metamotif is in fact simply a post-processing step done after aligning motif columns and cutting the motifs to a fixed length (Step 2 in Figure 2.3): a maximum likelihood Dirichlet distribution is computed using the Newton iteration method described in Minka (2003) due to the lack of a closed form solution. The motif set alignment algorithm which I implemented was also made to allow outputting an average PWM (a familial binding profile -like construct, see Section 2.1.1), or the alignment as a series of aligned motifs. All of these output options (a metamotif, an average motif, and an aligned set of motifs) are also available in iMotifs (Piipari et al., 2010b).

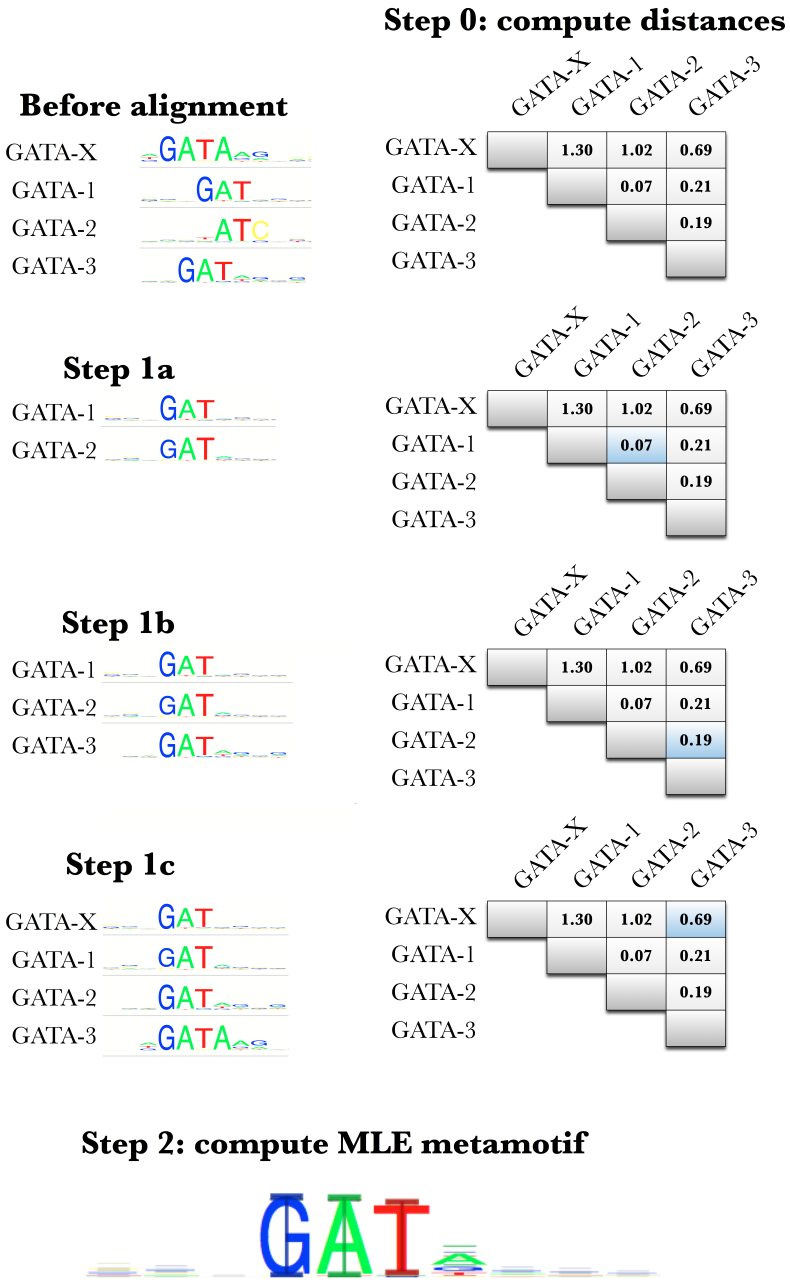


Figure 2.3: Schematic explaining the MLE metamotif inference algorithm. Firstly a distance matrix is computed between the motifs (Step 0). Motifs are added to the alignment in the order of increasing minimal pairwise distance to the motifs already in the alignment (steps 1a,1b,1c). Note that the motif GATA-2 is reverse-complemented upon adding it to the alignment. Motif ends are optionally cut such as to arrive at a motif alignment with no 'hanging end columns' (a minimum number of motifs with a supporting column can be defined to choose the threshold). A MLE Dirichlet distribution is then estimated for all motif columns using the method described in Minka (2003).

2.2.4 Metamotif inference by nested sampling

The metamotif can be seen as a way to summarise a gapless alignment of motifs of a certain length, to yield a probability distribution of motifs. However, my goal in designing the metamotif framework was to describe recurring patterns seen in sequence motif data deposited in public motif databases such as TRANSFAC (Matys et al., 2006), JASPAR (Portales-Casamar et al., 2010) or UniPROBE (Newburger and Bulyk, 2009). Many sequence motif families cannot be described accurately by global gapless multiple alignments of motifs at a fixed length. Motifs can for example consist of shorter repetitive signals, such as in the case of the heat-shock factor (HSF) motifs (Figure 2.4D), or the basic Helix-Loop-Helix (bHLH) motif family that are completely or partially palindromic due to their dimeric binding mode (Anthony-Cahill et al., 1992). Inspection of the HSF motif set shows that a global alignment of its columns does not describe the regularly spaced five-base repeat that is observed as part of the motifs in opposing orientations (aGAAn / nTTCt) (Kroeger and Morimoto, 1994). Furthermore, even non-repetitive and non-palindromic motifs present challenges for gapless multiple alignments: the span of informative columns contributing to familial patterns in publicly available PWM data is often unclear because of different signal-to-noise ratios and varying information content criteria used for calling motif ends.

I wanted to develop an inference algorithm that allows simultaneous detection of n short metamotif signals from a set of motif data, allowing for varying length for different metamotifs, and optionally free orientation (signal present on either strand). The metamotif count n is a fixed, user settable parameter to the algorithm. For metamotif inference problems where n is expected to be large, the choice for the parameter should be informed by prior information of the motif set under study, for example clustering of the motifs to estimate a rough number of recurring motif segments. Each metamotif has *a priori* an unknown length between l_{min} and l_{max} columns, and is expected to contain one or more matches in a fraction f of motifs. Motifs in the framework are thought to be generated by recurring metamotif patterns, each of which is potentially shorter than any of the motifs, and background positions that model "uninteresting" sections of the motifs (positions not emitted by any of the metamotifs). The background

model in the framework is the maximum likelihood (MLE) Dirichlet distribution estimated from all the motif columns in the input data. It is computed with the optimisation procedure described in Minka (2003), which is also used in the simpler MLE metamotif inference algorithm described in Section 2.2.3.

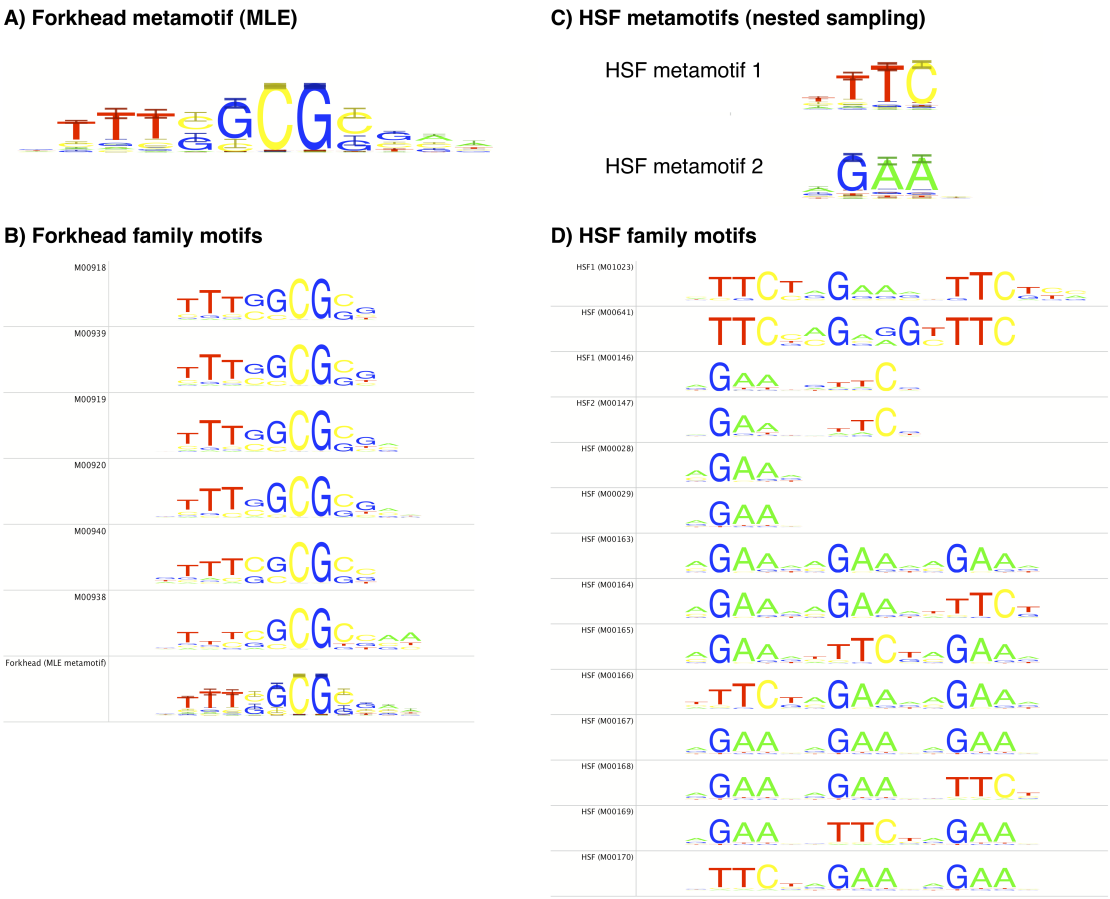


Figure 2.4: Example metamotifs for forkhead (A) and HSF (B) motif families from the TRANSFAC database (Matys et al., 2006). A) The MLE metamotif estimated for a subset of forkhead motifs (B) in the TRANSFAC 12.2 (Matys et al., 2006) regulatory motif database. C) Two HSF metamotifs estimated using the metamotif nested sampling algorithm from a subset of HSF motifs (D) in the TRANSFAC regulatory motif database.

The metamotif inference algorithm which I developed is a variant of the NestedMICA nested sampling algorithm described in Section 1.3.3. Nested sampling, originally introduced by Skilling (2004), is a generic Bayesian MCMC sampling strategy that allows drawing samples from a posterior distribution and directly estimating the evidence (marginal likelihood) of the model.

The metamotif nested sampler takes recurring intra-motif structure into account and allows detection of multiple metamotifs from a set of motifs. Motif sets are treated as a combination of short recurring patterns emitted by metamotifs, and background positions. The recurring signal can also optionally be allowed to be present on either strand, further improving the ability to detect repeating features. Recurring metamotif signals of interest are modelled separately from the "uninteresting" sections of the motifs that are taken as having been generated by a background model. The background model is the maximum likelihood (MLE) Dirichlet distribution estimated from all the motif columns in the input data. It is computed with the optimisation procedure described in Minka (2003), which is also used in the simpler MLE metamotif inference algorithm described in Section 2.2.3.

The algorithm allows estimating n metamotifs for a set of p motifs, with a variable metamotif length between l_{min} and l_{max} columns, and an expected fraction f of motifs containing any one of the n metamotifs. This is analogous to the NestedMICA motif inference algorithm that estimates multiple motifs with varying length from an expected fraction of nucleotide or protein sequence data. The posterior distribution being sampled is over the sets of n metamotifs and so-called mixture matrices, given the motif data and a background model for the motifs. The mixture (or occupancy) matrix describes the pairing between metamotifs and motifs. The term mixing matrix is a reference to the algorithm treating pattern recognition as an independent component analysis problem similar to the NestedMICA motif inference algorithm (Section 1.3.3): a likelihood function is written for the observations (the motif set) and the motif set is assumed to be generated as a mixture of independent metamotif contributions and noise represented by the background model. Each element $\mathbf{Q}_{i,j}$ in the $n \times p$ mixing matrix $\mathbf{Q}$ is a binary indicator of the metamotif j being present one or more times in the motif i. If the metamotif is present, $\mathbf{Q}_{i,j} = 1$, otherwise $\mathbf{Q}_{i,j} = 0$. The likelihood of

the motif set given the metamotif set is simply the product of likelihoods of each individual motif given the metamotif set and the mixture matrix.

2.2.5 The likelihood function

The likelihood of a motif given a set of metamotifs is calculated assuming the motif is emitted by the multiple-uncounted motif–metamotif mixture model (a MUMM with two metamotifs is given in Figure 2.5). This formulation allows for each motif to contain multiple metamotifs simultaneously, without the need to iteratively repeat sampling after masking previously inferred stronger signals.

Computing the likelihood of a motif given metamotifs under the MUMM model involves completing one-dimensional dynamic programming from the beginning of the motif to column c, closely in the same form as the protein or nucleotide sequence likelihood function described for the NestedMICA algorithm in Dogruel et al. (2008) (Equation 2.8).

$$L_c = (1-t)B_{c-1}L_{c-1} + \frac{t}{|M|}\sum_{\alpha \in M} \mathbb{P}(\mathbf{X}_{c-l_\alpha+1}^{c-1})L_{c-l_\alpha} \tag{2.8}$$

L_c represents the likelihood of all metamotif and background column arrangements (paths) in the input motif up to the column c. M is the set of metamotifs that have a mixing coefficient of 1 for the motif under consideration (i.e. metamotifs marked to be present in the motif in the mixing matrix $\mathbf{Q}$), and $|M|$ is the number of metamotifs that have a mixing coefficient 1. The length of the metamotif α is represented by l_α. B_c is the probability that the motif column at position c was emitted by the background. For the motif $\mathbf{X}$ of length $l_\mathbf{X}$ the transition probability t to a metamotif is defined as $t = 1/l_\mathbf{X}$, i.e. one metamotif is expected per motif, and any motif position is equally likely to contain a transition. $\mathbb{P}(\mathbf{X}_i^j)$ is the probability that the motif segment from i to j was emitted by a metamotif m, and it is given by the metamotif density function (Equation 2.5). A metamotif can optionally be allowed to be present on either strand to improve the ability to detect repeating (e.g. palindromic) features. Alternating orientation of metamotifs are achieved simply by summing the probability contributions $\mathbb{P}(\mathbf{X}_i^j)$ of the metamotif α and its reverse complement at all possible

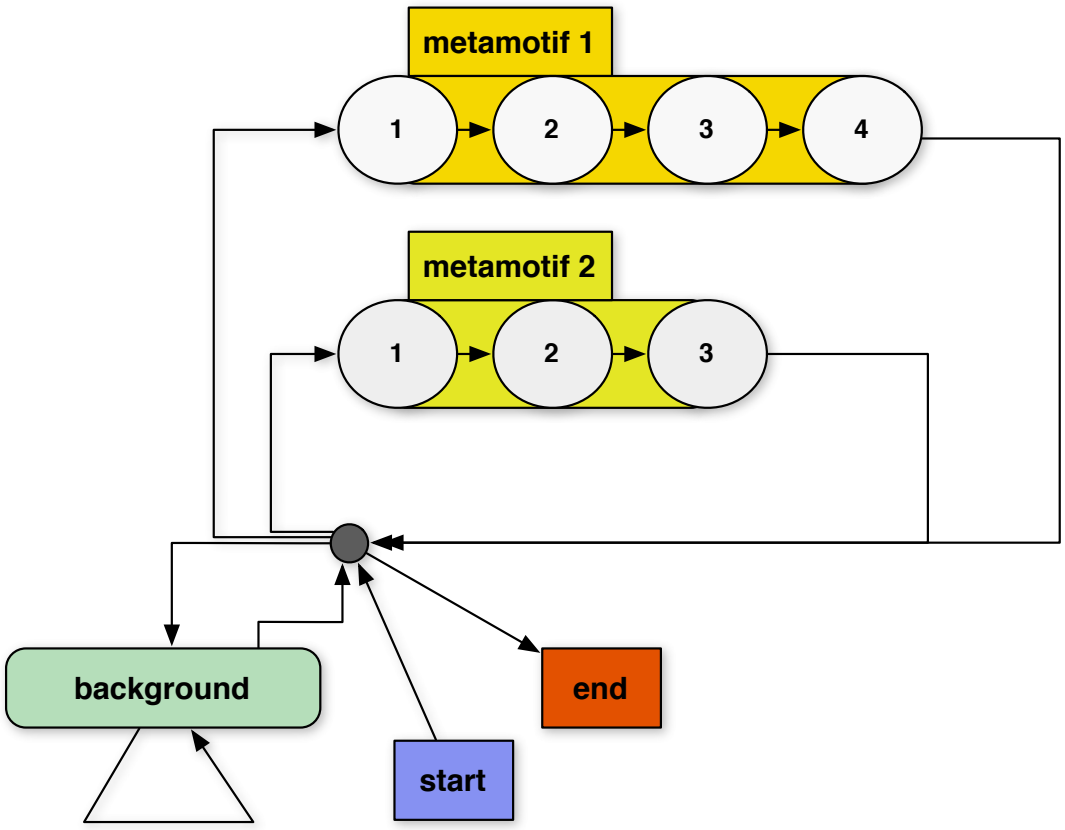


Figure 2.5: The multiple-uncounted motif–metamotif mixture HMM (MUMM). Numbered steps model the columns of the metamotif signals of interest and the background states are responsible for the "uninteresting" positions. Motif columns are emitted from a selection of metamotifs of varying lengths, and background positions. Note the similarity to the sequence–motif mixture model used in the NestedMICA motif discovery algorithm for motifs embedded in sequence (Figure 1.6).

offsets. Incomplete metamotif hits are also accounted for (Section 2.2.7).

2.2.6 Monte Carlo sampling moves

The metamotif nested sampler algorithm evolves metamotif parameters, and the mixture matrix state, with Monte Carlo sampling moves. Most of the proposal types alter the metamotif column parameters. The metamotif proposals are selected randomly from amongst the following set of moves:

- a small perturbation is made to a randomly selected metamotif column nucleotide mean weight: perturbation is made according to a randomly chosen nucleotide α weight α_i, nucleotide mean weights adjusted so they again sum to 1, and α_i of the column adjusted accordingly, maintaining precision unchanged.
- a small perturbation is made to a randomly selected metamotif column precision α_0: α_0 is perturbed, and α adjusted such as to maintain the mean nucleotide weights unchanged with a new precision.
- a small perturbation is made to a randomly selected metamotif column nucleotide weight α_i, thereby indirectly changing the precision.
- replacing a metamotif column with a new one, sampled from an uninformative simplex prior (nucleotide weights on the range [0.1, 40.0] are allowed).
- removing a column in one end of a metamotif while adding another one to the other end.
- adjusting motif length, by adding or removing a column from either end.

The two update operations that use an alternative parameterisation of α with precision and the mean nucleotide weights, i.e. updating the precision whilst maintaining mean weights unchanged, and altering the mean weights whilst maintaining the precision unchanged, proved beneficial for achieving convergence of the algorithm. When these moves were included, the algorithm converged consistently with smaller number of iterations than when only the more naive method

of updating α_i with random perturbations was included (data not shown). The prior function over the Dirichlet distribution parameters was an uninformative 'clipped' simplex prior: all values for the nucleotide weight parameters α_i of the distribution are allowed on the range $[0.1, 40.0]$ and equally likely. Parameter values above or below this range are clipped such as to avoid numerical instability.

Sampling moves are also done in the space of mixture matrices by flipping states of randomly selected elements in the mixture matrix similarly as done in Dogruel et al. (2008) for the NestedMICA algorithm.

2.2.7 Accounting for incomplete metamotif hits

Accounting for incomplete metamotif matches in a motif is an important consideration. This is because we wish to analyse data from different experimental and computational sources where motif start or end positions have not been chosen consistently, for instance with an information content criterion. Incomplete hits are accounted for by adding additional "un-informative" columns in the input motifs in both the $5'$ and the $3'$ motif ends. The un-informative columns are multinomial distributions that match the mean nucleotide weights of the background model Dirichlet distribution. This effectively allows all possible offsets of the metamotif that overlap the motif with at least one column, whilst associating more uncertainty to those columns supported by only a subset of the motif data (Figure 2.6).

2.3 Evaluating the metamotif nested sampler algorithm

Performance of the metamotif inference algorithm was tested using synthetic motif sets where samples from metamotifs were inserted, or "spiked", similarly as done by Dogruel (2008); Tang et al. (2008) with synthetic sequences and samples from motifs. The aim was to measure the relative frequency of metamotifs at which the expected metamotifs could be recovered by the algorithm from synthetic motif data containing metamotif instances. The evaluations were done in two stages. The ability of the algorithm to infer a single metamotif presented to

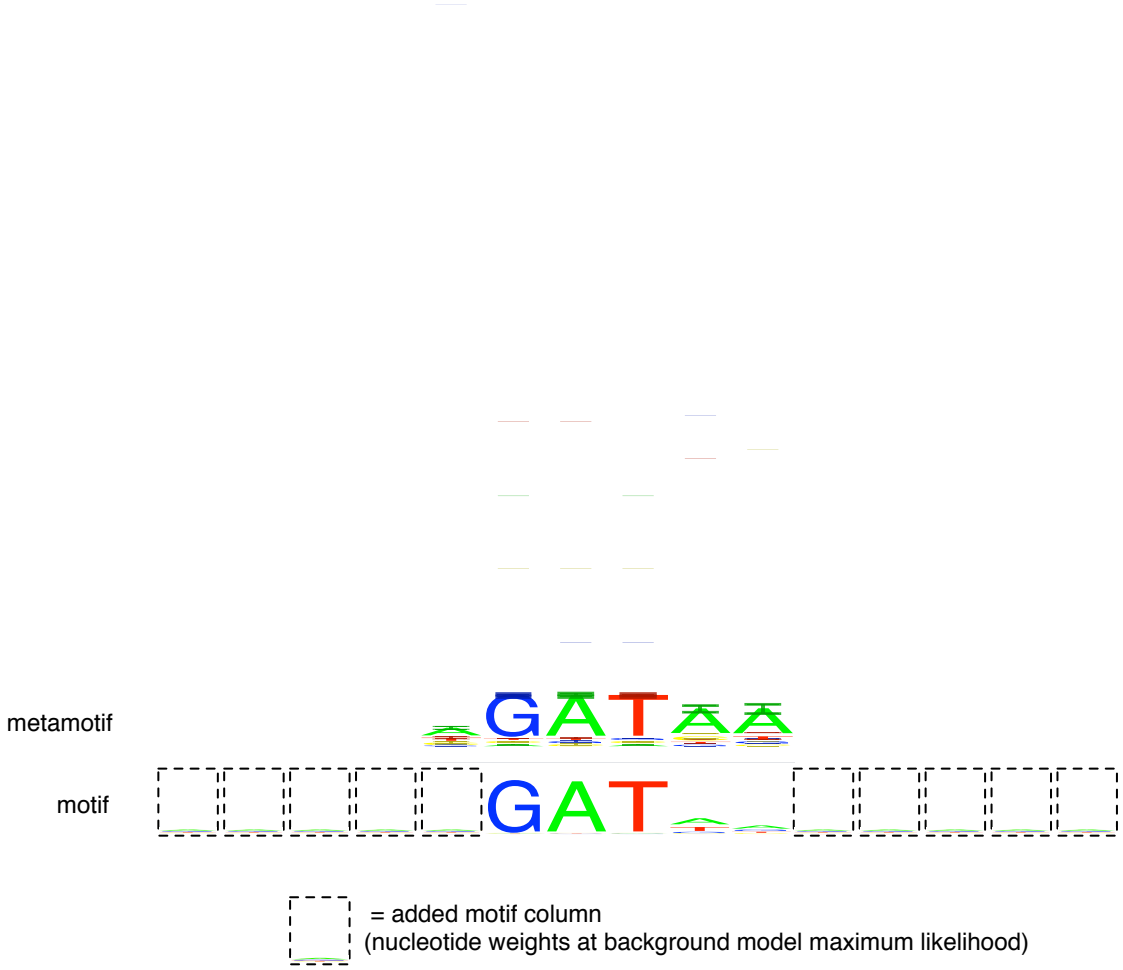


Figure 2.6: Incomplete hits are handled by padding the input motifs with additional columns that fit the background model optimally. All metamotif hits are required to be at minimum two columns long, which means that all input motifs are (optionally) capped with $l_{min} - 1$ additional columns, where l_{min} is the user settable minimum metamotif length parameter (which also has a minimum allowed value of 2).

it was tested first (Section 2.3.1). After that, several metamotifs were presented to the algorithm to assess the ability of the algorithm to infer multiple metamotifs simultaneously (Section 2.3.2). Metamotifs were then also inferred from the TRANSFAC database (Section 2.3.3).

To prepare the synthetic motif sets, metamotifs were first generated of examples of three structurally diverse TRANSFAC 12.2 PWM families: six forkhead motifs (class 3.3 in TRANSFAC classification), six GATA-like Cys_4 zinc finger motifs (class 2.1) and five MADS box motifs (class 4.4) were used (source motifs shown in Figure 2.7). This was done by aligning each of the three input motif sets with a greedy gapless sequence motif multiple alignment method related to the one utilised in STAMP motif toolkit (Mahony and Benos, 2007). A metamotif was then estimated from the motif multiple alignments with the MLE method from Minka (2003): MLE Dirichlet distribution was computed for motif alignment columns (example seen in Figure 2.4A), with each motif column in the alignment mapping to a MLE Dirichlet distribution in the resulting metamotif.

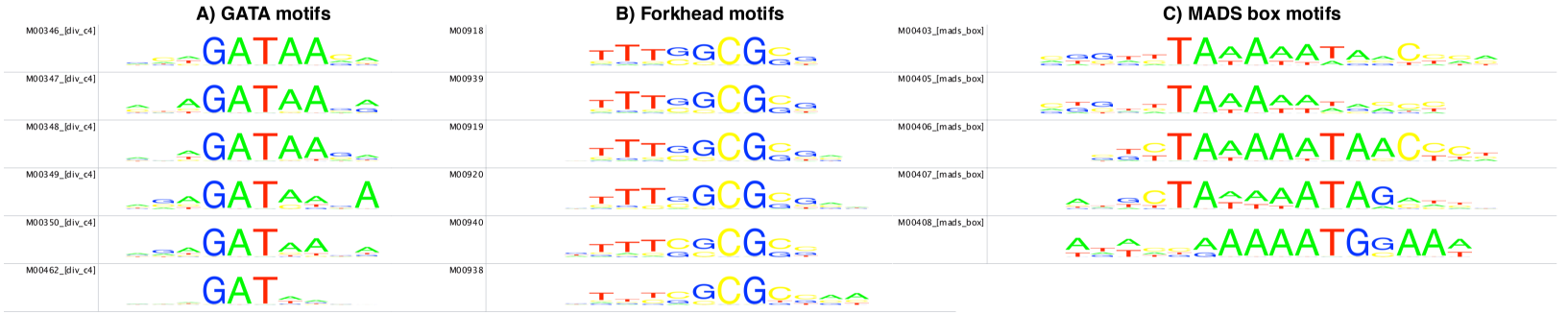


Figure 2.7: These motifs were aligned and the multiple alignment summarised as an MLE metamotif with the program *nmalign*. See the topmost metamotifs in Figure 2.8 for the resulting target metamotifs that were spiked into synthetic motif sets.

Motifs (PWMs) from each of the three familial metamotifs were sampled in relative frequencies of 0%, 10%, 20%, ..., 100%, into synthetic input motif sets (separate input motif set per motif family). Each synthetic motif set contained 60 motifs, each 20 nucleotide columns long, with a maximum of one metamotif instance allowed per input motif. The synthetic motif columns in the input motif sets are samples from a Dirichlet distribution with parameters $\alpha = \{0.5, 0.5, 0.5, 0.5\}$. The metamotif sample PWMs were inserted at random positions within the 20 nucleotide long synthetic motifs. The metamotif inference algorithm was then run on the motif set to infer a single metamotif between length ranges 4 and 14, allowing for the signal to be present in either orientation (`-numMetamotifs 1 -revComp -minLength 4 -maxLength 14`).

Metamotif inference performance was measured qualitatively with visual inspection comparing the inferred metamotifs to the known spiked metamotifs, and quantitively measuring the Cartesian distance between the metamotif mean nucleotide weights.

2.3.1 A single metamotif

The metamotif nested sampler algorithm was used to infer metamotifs from the synthetic motif sets to evaluate how well the spiked metamotif patterns could be recovered. Performance was measured qualitatively with visual inspection comparing the inferred metamotifs to the known spiked metamotifs, and quantitively measuring the Cartesian distance between the metamotif mean nucleotide weights. The visual comparison, Cartesian distances and empirical p-values for observed metamotif-metamotif distances are presented in Figure 2.8. The evaluation shows that metamotifs can be inferred from motif sets that contain them with relative frequencies of even 10%. At a relative frequency of 40% and above all three recovered metamotifs are very similar to the respective source metamotif (Figure 2.8).

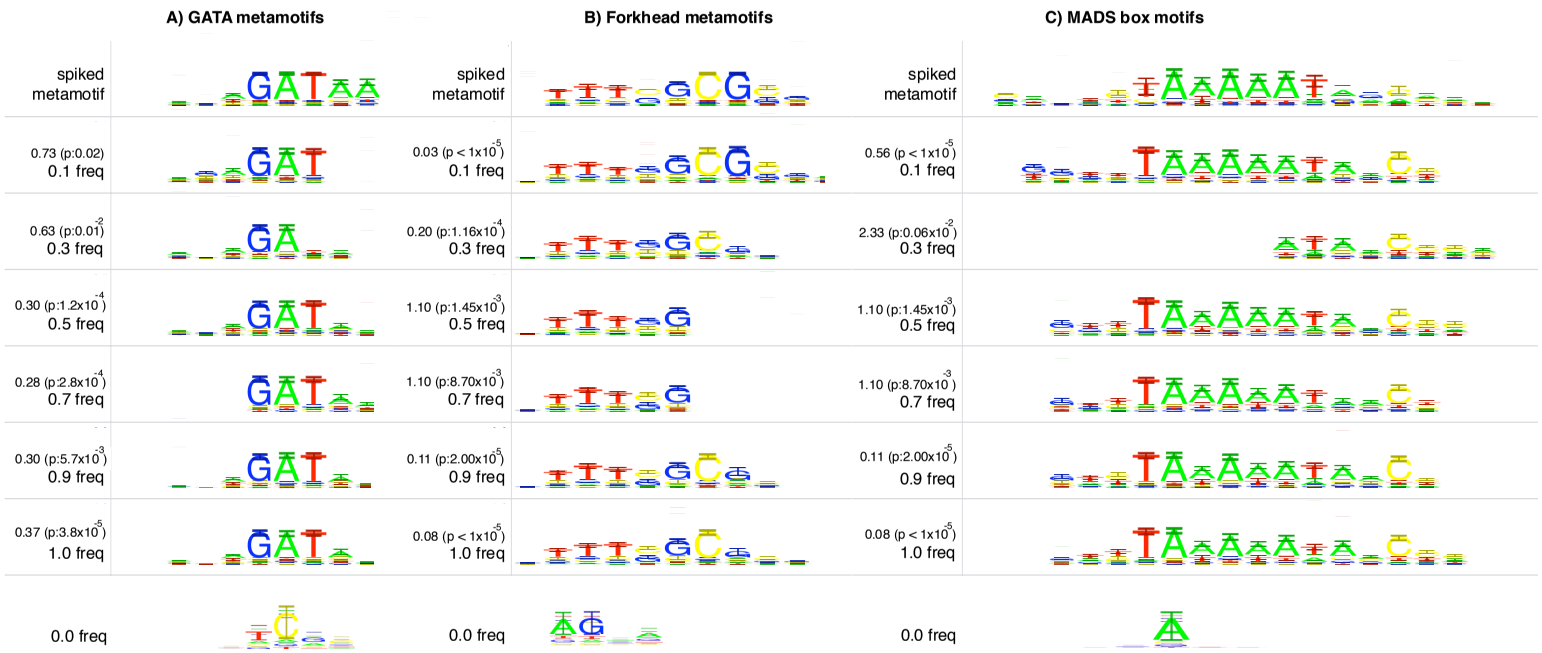


Figure 2.8: Metamotifs estimated with the metamotif nested sampler algorithm with varying relative frequency of metamotif samples. The top row in each metamotif alignment contains the "correct" metamotif that was sampled to the input weight matrix data in six different relative frequencies: 0.0, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0. Frequency 0.0 which is shown in the bottom of the graph refers to a control experiment where all columns of the motif set are samples from the background (a Dirichlet distribution with parameters $\alpha = \{0.5, 0.5, 0.5, 0.5\}$). A Cartesian-like distance between the sampled metamotif column mean nucleotide weights of the shown metamotif and the spiked metamotif mean nucleotide weights is presented above the relative frequency. An empirical p-value as described by (Down et al., 2007) is also shown for the Cartesian distances (100,000 shuffles made for each motif).

2.3.2 Multiple metamotifs

The ability to predict multiple metamotifs was demonstrated in a second evaluation experiment where instances of all the three motif families were inserted into synthetic motif sets and the algorithm was required to infer three metamotifs. It was shown that the algorithm was able to infer multiple metamotif models concurrently with correct lengths at a relative frequency as low as 20% (Figure 2.9).

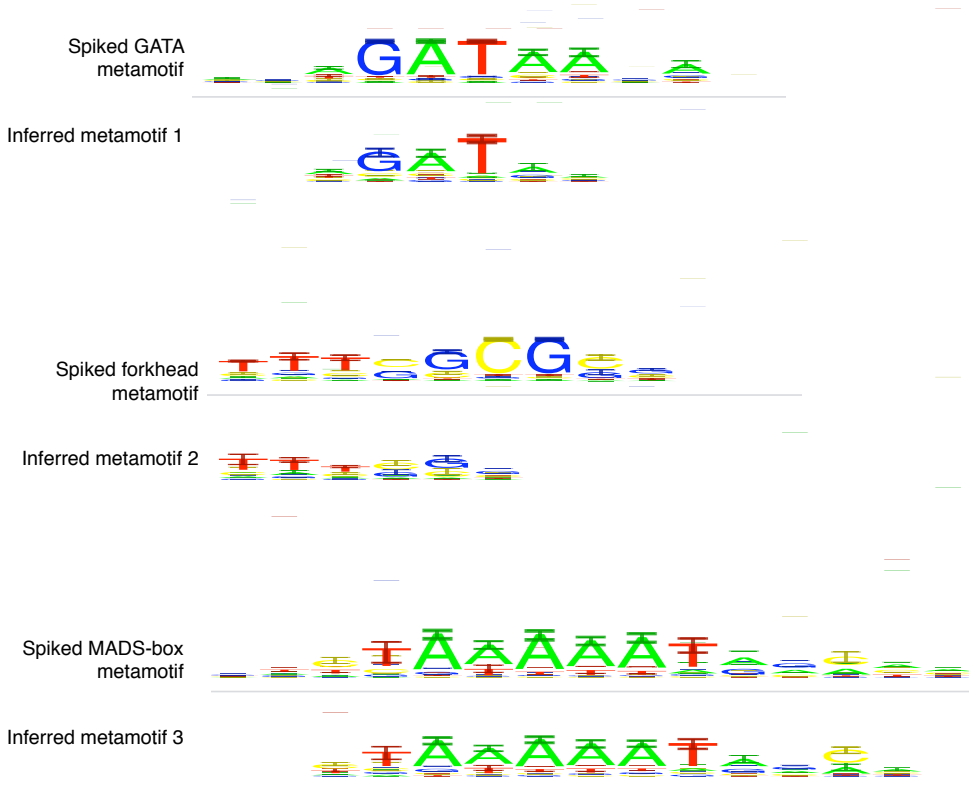


Figure 2.9: The metamotifs predicted at relative frequency of 0.2 are shown alongside the source metamotifs.

2.3.3 Inferring metamotifs from TRANSFAC

I demonstrated use of the metamotif nested sampling algorithm in inferring familial metamotifs from known experimentally determined regulatory motifs from the TRANSFAC database (Matys et al., 2006). Motifs retrieved from TRANSFAC were first divided to clusters with the SSD distance by Down et al. (2007) with cutoff 6.0. Three metamotifs were then inferred from each of the resulting clusters. Examples of metamotifs inferred are shown in Figure 2.10. The metamotif nested sampler algorithm was found capable of detecting several recurring patterns from the motif clusters that are clear upon visual inspection of the motifs, in addition to finding overliers from the motif sets (Figure **??**B).

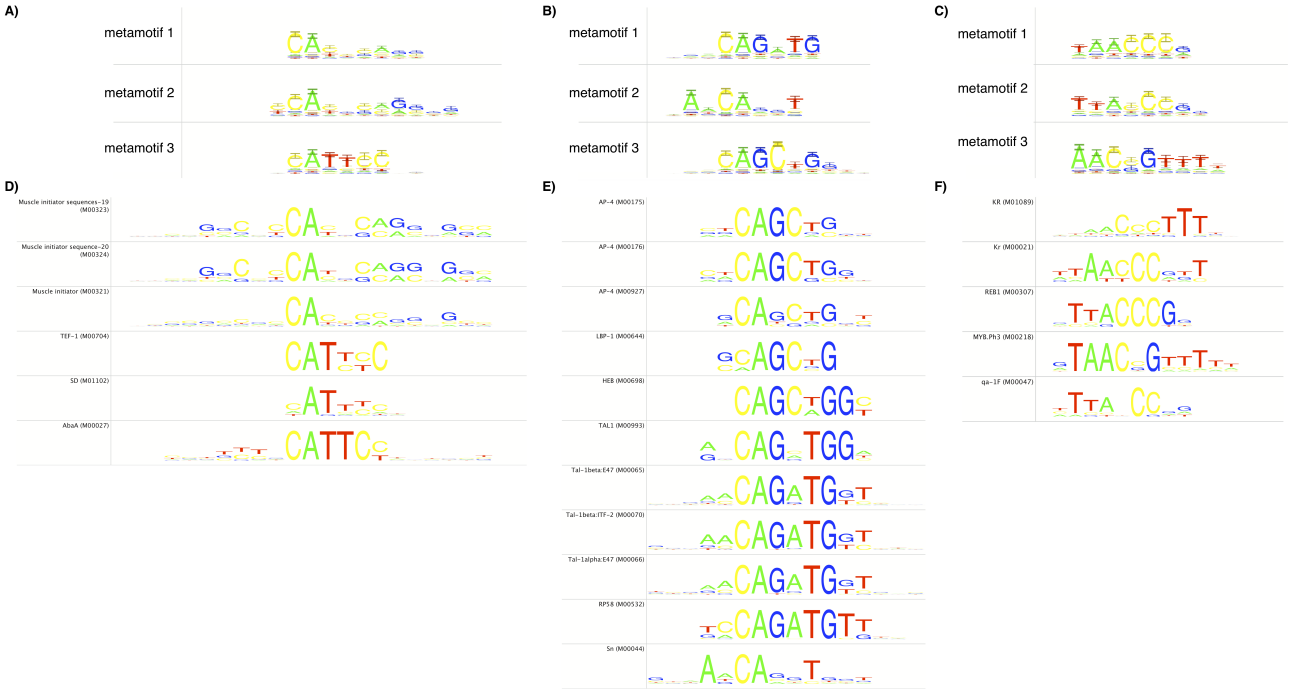


Figure 2.10: Examples of metamotifs inferred with the nested sampler algorithm from clustered motifs deposited in the TRANSFAC database (Matys et al., 2006). Three metamotif sets (A,B,C) were estimated from input motif sets (D, E and F, respectively). A) Two largely similar patterns (metamotifs 1 and 2) are inferred that correspond to the muscle initiator like motifs, with one distinct pattern (metamotif 3) corresponding to the CATTCC-like motifs (members of the TEA family) also found from the cluster. B) shows that the metamotif nested sampler distinguishes a CAGATG-like (metamotif 1) and CAGCTG-like pattern (metamotif 2) from the motif cluster E, and separates an Sn (M00044)-like outlier from the two patterns as metamotif 3. C) Three metamotif patterns inferred from the motif set separate the TAACCCG-like (metamotif 1), TTACCCG (metamotif 2) and TAACCGTTT (metamotif 3) like patterns seen in the motif cluster. Notably metamotif 3 contains a shorter CCG-like core instead of the CCCG found in the two other metamotifs.

Evaluation of the nested sampling based metamotif inference algorithm suggests that it is able to correctly infer familial metamotif patterns. It performs both in the case of a single recurring motif family, and in the case of motif sets with examples of multiple motif families. This makes it potentially applicable for instance for finding redundant motif patterns from large scale *de novo* inferred sets of motif predictions from different algorithms, or for inferring a complete set of familial metamotifs from a set of motifs. Metamotif inference is also conducted from clustered motifs from the TRANSFAC database.

2.4 Summary

In this chapter I introduce a generative model for PWM motif columns, called the metamotif. The metamotif is a a probability distribution over PWM motifs of a given length. I also present a nested sampling based algorithm for inferring metamotif parameters from a set of motifs.

All of the following chapters make use of the metamotif in one way or another: Chapter 3 introduces a variant of the NestedMICA motif inference algorithm with an informative motif prior based on the metamotif likelihood function (Equation 2.5). Chapter 4 presents a motif family classification method based around metamotifs. In Chapter 5 I then experiment with using the metamotif based classification method with *de novo* discovered motifs.