

Chapter 2

NestedMICA as an *ab initio* protein motif finder

1

2.1 Background

Discovering linear sequence motifs common to a set of protein sequences has long been an important problem in biology. It is possible to check if a set of proteins contain a known sequence motif by searching protein motif or domain databases. Databases including Pfam (Bateman *et al.*, 2004), eukaryotic linear motif database (ELM) (Puntervoll *et al.*, 2003), Prosite (Hulo *et al.*, 2006) and ScanSite (Obenauer *et al.*, 2003) contain sequence motifs and domains in the form of regular expressions or profile HMMs. Obviously, one cannot use these resources to discover a novel or unannotated sequence motif that is suspected to be a common feature in a given protein set. While new protein domains such as

¹This chapter was partly published in BMC Bioinformatics in January 2008 (Doğruel *et al.*, 2008), by the author of this PhD thesis (MD), Dr. Thomas Down (TD), and finally my thesis supervisor Dr. Tim Hubbard (TH). Authors' contributions are as follows: TH and MD conceived this work, MD and TD modified the NestedMICA code, MD performed the tests and wrote the manuscript.

2.1 Background

those contained in Pfam can be defined from alignments of evolutionarily related sequences, the identification of short sequence motifs, potentially shared between proteins that appear evolutionarily unrelated, is much harder.

To tackle this problem, several multiple alignment approaches (Hertz & Stormo, 1999; Zaslavsky & Singh, 2006) have been proposed. One such tool, Dilimot (Neduva & Russell, 2006), is a recent protein motif search tool aiming at finding relatively short overrepresented motifs by aligning only sequence regions that are likely to contain a linear motif. It filters out regions including globular domains and coiled-coil regions which are reported or predicted by some other algorithm, before searching for known motifs in several protein databases such as PFAM, and finally uses a pattern search program, TEIRESIAS (Rigoutsos & Floratos, 1998) to find overrepresented matches. TEIRESIAS, an *ab initio* program that is not based on database look-up, can list frequently repeating character-based patterns that include gaps, from a given sequence set. Patterns can include one or two events separated by wild-card characters, as in AT..G (Burgard *et al.*, 2001). Another similar and robust amino acid pattern search tool is SLIMFinder (Edwards *et al.*, 2007) in which short protein motifs are built by combining dimers into longer patterns, retaining only those motifs occurring in a sufficient number of unrelated proteins. Motifs with fixed amino acid positions are identified and then combined to incorporate amino acid ambiguity and variable-length wildcard spacers. Dilimot, TEIRESIAS and SLIMFinder report results as regular expressions. There are also other algorithms in the non *ab initio* motif finding category, using evolutionary or structural information, which are specifically designed to predict DNA-binding regions in protein sequences (Ahmad & Sarai, 2005; Hwang

2.1 Background

et al., 2007; Kuznetsov *et al.*, 2006). However since the MEME tool was developed (Bailey & Elkan, 1995) and provided a way to carry out *ab initio* protein motif finding, returning a set of Position Weight Matrices (PWMs) rather than regular expressions, not many multi-purpose sequence-based probabilistic motif finders have been developed, despite there being numerous tools for finding motifs in DNA. Examples to other well known DNA motif discovery tools are SeSiMCMC (Favorov *et al.*, 2005), AlignACE (Hughes *et al.*, 2000), ANN-Spec Workman & Stormo (2000), Weeder (Pavesi *et al.*, 2004), and YMF (Sinha & Tompa, 2003).

NestedMICA (Down & Hubbard, 2005) is a probabilistic motif discovery algorithm which uses a new Monte Carlo inference strategy called Nested Sampling (Skilling, 2004). Written in the Java programming language as an open source application, NestedMICA uses Biojava libraries (BioJava, 2007). It has been successfully used for transcription binding site and large-scale promoter motif discovery (Down & Hubbard, 2002). In this manuscript, I extend the application of NestedMICA to finding motifs in protein sequences and compared it with the popular program MEME using both biologically-authentic and synthetic test data sets. I chose to compare NestedMICA with MEME, because the output of MEME is motifs in the form of PWMs, making comparison possible. MEME is also an *ab initio* method and uses probabilistic models like NestedMICA.

To evaluate the performance of the two methods I have performed various spiking tests in which some test motifs generated from protein domain alignments were spiked into a set of protein sequences, as described in the Methods. This assessment procedure is similar to the approach followed in a previous transcription binding site motif discovery programs comparison by Tompa *et al.* (2005).

NestedMICA has also been assessed by testing its ability to find a subcellular localisation motif in datasets known to contain a specific localisation signal.

2.2 Materials and methods

2.2.1 NestedMICA

NestedMICA is a probabilistic motif inference method based on a generative sequence model. The model has three sets of parameters: firstly, a background model which represents all the non-motif parts of the input sequences; second, a set of position-weight matrices which represent the motifs themselves; finally, a binary matrix (the occupancy matrix) whose elements specify whether a given motif should be considered when modeling a given input sequence. The background model is built in advance and held constant during motif inference, while the motifs and occupancy matrix are updated to fit the supplied data. NestedMICA uses the Nested Sampling strategy (Skilling, 2004) to update both of these sets of parameters.

The implementation of NestedMICA's `nminfer` program can be split into two major parts: code that calculates the likelihood of some sequences under the generative model, and code which implements the Nested Sampling process. The Nested Sampling code makes few assumptions about the internal structure of the model (and could potentially be used to perform inference of quite different models), so I consider these two components separately.

NestedMICA was designed completely in an object oriented and modular manner that allows one to plug in a very different model without touching the trainer code: Similarly, the likelihood calculators do not know anything about Nested

Sampling (and could potentially be used in another training framework). Below, sequence models, likelihood calculation, nested sampling, and finally the implementation are discussed.

2.2.1.1 The NestedMICA sequence model

NestedMICA relaxes the constraints of the ZOOPS model (see 1.3.2) slightly by allowing a given motif to appear multiple times in the same input sequence. To calculate the likelihood of a given sequence, NestedMICA first consults to appropriate row of the occupancy matrix to determine a (possibly empty) subset, M , of the complete motif set which applies to this sequence. In the case where M is empty, the likelihood of the sequence is simply its likelihood under the background model (see below). When M is non-empty, NestedMICA sums over all possible configurations of motif occurrences along the sequence, filling in any gaps using the background model. This is performed using a dynamic programming recursion which gives the likelihood, L_n of all paths up to a given point in the input sequence, n as:

$$L_n = (1 - t)B_{n-1}L_{n-1} + \frac{t}{|M|} \sum_{m \in M} m(S_{n-|m|+1}^{n-1})L_{n-|m|} \quad (2.1)$$

where $|M|$ is the number of motifs selected by the occupancy matrix, $|m|$ is the length of weight matrix m , B_n is the probability that the sequence symbol at position n was emitted by the background model, $m(S_i^j)$ is the probability that the sequence from i to j was emitted by the weight matrix m , and t is a transition probability specifying the estimated density of motifs in the sequence.

We initialise $L_0 = 1$ then apply the above formula recursively along the length

of the input sequence until the final position is reached, giving a likelihood for the complete sequence.

In principle, any background model could be used with this formulation. In practise, I choose to use a mosaic background (Down & Hubbard, 2005) which admits the possibility of several different classes of background sequence, each of which is modeled using a low-order Markov chain (*i.e.* within a given class, the probability of observing a particular symbol at position n depends on the symbols observed at a fixed number of previous positions). The mosaic model is implemented as a fully connected HMM (transitions are allowed between any pair of classes).

To calculate B_n , NestedMICA first applies the standard posterior decoding algorithm (Durbin *et al.*, 1999) to find P_{hn} , the posterior probability that the symbol at position n in the input sequence was generated by state h of the background model H . We can then calculate B_n as:

$$B_n = \sum_{h \in H} P_{hn} h(S_n) \tag{2.2}$$

(*i.e.* summing over any remaining uncertainty in which background class is used at n). Note that when the Markov chain order, o is greater than zero, the probability of observing a given symbol, $h(S_n)$, depends on o previous symbols in the sequence. This means that it is not possible to exactly calculate B_n where $n \leq o$. We choose to ignore the first o symbols in the input sequence (except for background calculation purposes) in order to avoid any edge effects.

2.2.1.2 Implementation of NestedMICA

The NestedMICA `nminfer` program is based around a fairly general implementation of the Nested Sampling strategy, which can be applied to any probabilistic model. This code takes three inputs: a data set (*i.e.* a set of sequences), some code to calculate the likelihood of the dataset given a model state (*i.e.* an implementation of the likelihood function given above), plus a set of “sampling” operations which perturb a state and can be used to move around state space.

Each state consists of two sets of parameters: a set of motif weight matrices, and an occupancy matrix specifying whether the motifs appear in the input sequence set. Most of NestedMICA’s sampling moves are applied to one randomly selected weight matrix (WM):

- making a small perturbation to one column of a weight matrix, by slightly increasing or decreasing one of the weights, then renormalizing so they still sum to 1.
- replacing a WM column with a new one, sampled from the prior.
- removing a column in one end of a WM while adding another one to the other end.
- adjusting motif length, by adding or removing a column from either end.

In addition, it is necessary to resample the occupancy matrix. In principle, a straightforward and valid sampling move would be to simply flip the state of one randomly-selected element in the occupancy matrix. In practise, NestedMICA

tests multiple occupancy matrix moves at the same time, since this improves performance when running on multi-processor systems.

Finally, it is necessary to place a prior over the state space. NestedMICA uses a simple non-informative prior for the Weight Matrix motif models: a uniform prior over weight-matrix space with a constraint that extremely low weights are forbidden. The lower limit is specified by the `-minClip` parameter and is typically 10^{-7} for amino acid, and of the order of 10^{-3} for dna input. We also place a non-informative prior on the occupancy matrix, although if there is some prior knowledge about the frequency of the target motif in the dataset, this can be specified using the `-expectedUsageFraction` option.

The main challenge when implementing nested samplers is to sample uniformly from the prior while respecting the likelihood constraint. In practice, this is usually solved by duplicating a randomly-selected state from the ensemble then using classical (single-state) Monte Carlo strategies to move the duplicate state. NestedMICA uses a straightforward Metropolis-Hastings approach for prior sampling. Further information on the use of this strategy is available in the original publication of NestedMICA ([Down & Hubbard, 2005](#)).

Rather than storing the weight matrix in its traditional form as a list of probability distributions over an alphabet, in NestedMICA it is stored as a circular buffer of distributions that is slightly larger than the longest motif being modeled, with the addition of an offset parameter (where the motif starts in the buffer) and a length parameter. The nice thing about this representation of motifs is that it is possible to extend the motif in either direction when length is needed to be sampled, up to the size of the circular buffer.

2.2.2 Adding protein support to NestedMICA

I made several changes to NestedMICA in order to support protein motif discovery. Firstly, I added support for loading and analysing protein sequences (enabled with the “-alphabet PROTEIN” switch). The inference strategy remains identical to that previously described (Down & Hubbard, 2005). However, the dimensionality of the protein motif discovery problem is much higher than in nucleic acids: a DNA motif model has three free parameters per position, while a protein motif has 19. To compensate for this difference, I found that a rather larger ensemble of models in the Nested Sampling process was required than for DNA. Having found an optimal ensemble size by performing a systematic parameter sweep test, I altered this to be the default ensemble size when running the program in protein mode. Unless set otherwise by the user, it is automatically set to either 4000 divided by number of target motifs, or set to a minimum of 1000, in case the division would be less than 1000.

Another important difference between the protein-capable version and the previous version of NestedMICA is the way distribution probability initialisation is performed in setting up the amino acid probability distributions for each background mosaic class. Starting off with flat probability distributions in all the mosaic classes of a given background as in the DNA case was not ideal for protein sequences, as I observed a minimal learning rate with these equal initial states. Instead, a semi random, semi actual input-based initialisation was preferred: the distributions were initialised such that they directly reflect the amino acid distributions of the actual input data, except, these numbers were slightly changed

randomly by a certain margin for the training to learn and converge faster.

Since the initial publication of NestedMICA (Down & Hubbard, 2005), an important extra feature was added of automatically optimising a motif's length within a user-specified motif length range. NestedMICA treats the motif length as another free parameter of the motif model, and optimises it using the same Nested Sampling strategy as for all the other parameters. Another change in the new version is that, if no background model is provided by the user, NestedMICA uses a basic, zero-order background model which is trained on the fly from the user supplied input sequences.

Further information regarding the parameters used in motif finding can be found in the user manual at the NestedMICA web site:

<http://www.sanger.ac.uk/Software/analysis/NestedMICA/>

2.2.3 Program output and sequence logos

NestedMICA reports discovered motifs as PWMs which can be viewed as sequence logos by an accompanying motif-viewer tool. In a single NestedMICA protein motif logo, each column has a maximum information content of 4.32 bits ($\log_2 20$), and amino acid letters are coloured according to their general physical and chemical properties, as depicted in Figure 2.2

As opposed to majority of motif finders, NestedMICA does not report any significance measures such as E-values, or entropy scores, as these values could be quite unreliable. All these scores are calculated based on the idea that a motif finder has picked up a real motif, which obviously cannot always be true. The recent publication by Ng *et al.* (2006), discusses in detail why using such scores

could lead to undesirable results.

2.2.4 Background model training

Probabilistic motif finding tools usually employ background models to represent sequence regions where ideally no motif of interest exists. In most cases, however, these programs use a homogenous background model, assuming that all non-motif portions of the sequence can be represented using a single amino acid frequency distribution. In reality, protein sequences are generally composed of different functional domains which can be chemically biased towards certain compositional forms. In addition, protein sequences are very likely to carry different sequence signals responsible for various molecule-recognition and binding related tasks. NestedMICA uses non-homogenous (“mosaic”) background models which subdivide the background sequences into several classes. Each class is modelled as a Markov chain. The order of the chain (i.e. the number of previous symbols on which the probability distribution for the next observed symbol is conditioned) can be set to an arbitrary value, but for protein sequence analysis I recommend only using zeroth or first-order background models, since higher order models will have an extremely high parameter count and will be hard, if not impossible, to parametrise effectively.

A built-in background likelihood estimation procedure in NestedMICA (called “nmevaluatebg”) allows an optimal background model architecture to be found for a given set of sequences. A NestedMICA background model can be of any order Markov chain and consist of an arbitrary number of mosaic classes. As a good representative sequence set, I used the pTarget protein subcellular locali-

sation dataset (Guda & Subramaniam, 2005) for background model parameter optimisation (Figure 2.1). This is mainly because it includes different types of proteins from different subcellular localisations, eliminating the chance of some strong domain and localisation signals to dominate the background model training and evaluation. Furthermore, I reduced the sequence identity of the set from 95% down to a maximum of 40% by using the CD-HIT (Li & Godzik, 2006) clustering software to have a total of 7437 eukaryotic proteins, which had an average sequence length of 522. For evaluation purposes, 6000 of these were used to train several different background models with different parameters, while the remaining sequences were used to test how well a certain background model represented them. As Figure 2.1 shows, using order-1 probabilities, where the compositional probability of a certain residue depends only on a single adjacent residue, performs better than a zero-order model. Moreover, likelihood for the test sequences increased monotonically with the number of mosaic classes. Training a multi-class higher-order background requires sufficient sequence data in order to prevent a possible over-fitting of the background. For example, using a first order, 6-classes model corresponds to having a total of 2400 different amino acid distributions.

2.2.5 Testing NestedMICA’s performance

In order to get a better understanding of NestedMICA’s protein motif finding capabilities and limits, a number of motif spiking experiments were performed using synthetic and biological motifs, similar to the approach previously used by Down & Hubbard (2005). In a motif spiking test, a number of short amino acid sequences are generated according to the weight matrix distribution probabilities

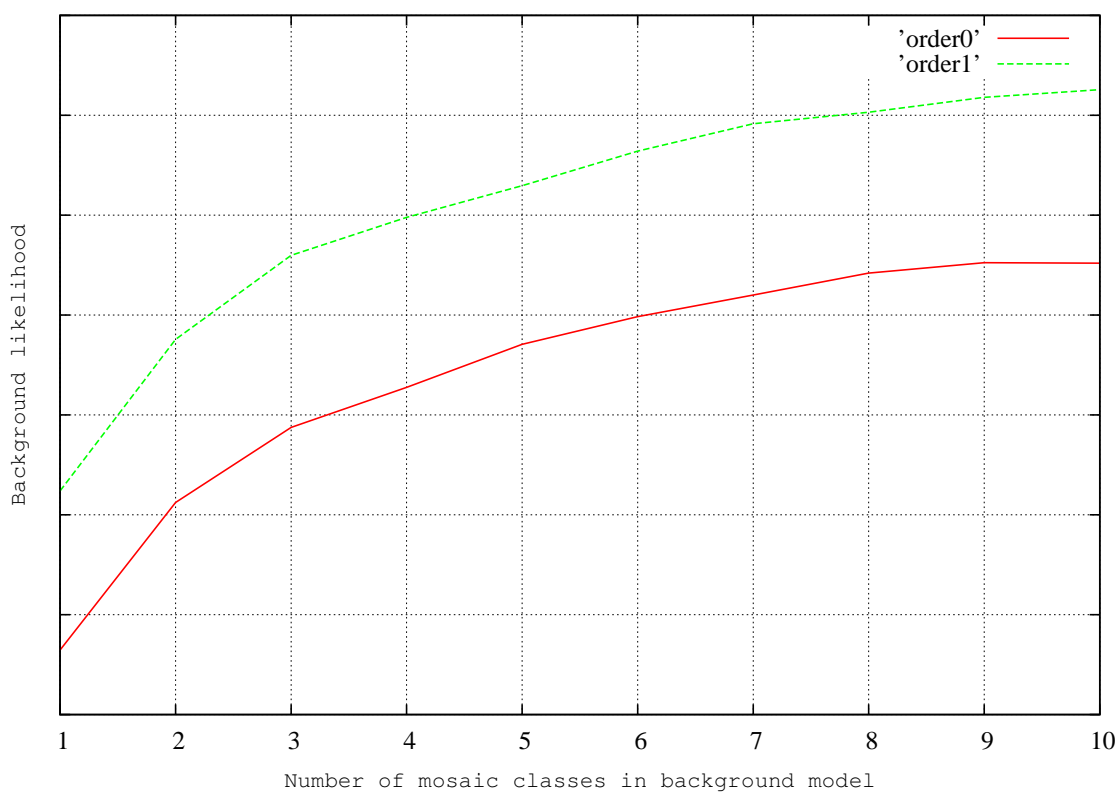


Figure 2.1: **Likelihood curve for different number of mosaic classes.** The x-axis represents the total number of mosaic classes in the tested background model architecture. The logarithmic y-axis corresponds to a likelihood measure that can take arbitrary values, of how well a background model represents the given sequence set. The red line represents a zero-order while the green one represents a first-order background model.

of a given motif. These motif-resembling short peptides are then inserted at random positions into a set of sequences. The program under test is then applied to the set of sequences to predict a set of motifs. Finally, the predicted candidate motif set is compared with the original test set to assess the performance of the program in recovering the spiked motifs. MEME PWMs were converted into NestedMICA sequence logos for easier comparison.

To evaluate how similar a reported motif is to the original one, I used Cartesian motif-motif distances. The Cartesian motif distance metric is the sum of individual Cartesian distances calculated for each motif position, between corresponding pairs of the 20 amino acid probabilities from both motifs. For a motif to be considered as recovered with a reasonable precision, I used an empirically set threshold for the maximum allowed Cartesian motif distance normalized for the original motif length. Motifs showing an average deviation per position of more than 0.3 of Cartesian motif distance were considered as false discoveries.

For each motif, in addition to reporting Cartesian motif distances, I calculated sensitivity (Equation 2.3) and specificity (Equation 2.4) values:

$$SN = \frac{TP}{TP + FN} \quad (2.3)$$

$$SP = \frac{TP}{TP + FP} \quad (2.4)$$

Matthew's Correlation Coefficient (MCC) (Matthews, 1975), shown in Equation 2.5, values were calculated, too, to show a PWM's scanning power as in Kiemer *et al.* (2005):

$$MCC = \frac{TP \quad TN - FN \quad FP}{\sqrt{(TN + FN)(TP + FN)(TN + FP)(TP + FP)}} \quad (2.5)$$

where TP, FP, FN, TN stand for true positives, false positives, false negatives and true negatives, respectively.

One advantage of using MCC in a PWM evaluation is that for random motif predictions MCC tends to be around zero, while for a perfect scanning performance it will have a maximum value of 1. On the other hand, depending on the choice of a score threshold, even for an irrelevant or weak motif one can get a sensitivity of 1, for instance, while the corresponding specificity value could be as low as 0.5, if the number of sequences in both datasets are equal. In such cases, MCC will tend to be very low, reflecting the random prediction.

To calculate these measures of motif scanning performance, first, I spiked every sequence in the test dataset with a particular motif, then I scanned a reported motif both in the spiked and original datasets to see how many motif instances would be correctly or falsely predicted in both datasets. For each individual test case, I picked a threshold score that maximises the corresponding MCC value, after trying a range of different score thresholds systematically incremented in each iteration to compute sensitivity, specificity and MCC values. I calculated these values not only for motifs reported by the programs I assessed, but also for the original test motifs. I did this because values measuring the scanning performances of recovered motifs should be considered relative to those of the original motif. A more objective and absolute metric of motif recovery is the Cartesian motif distance, which is the sum of probability differences in corre-

2.2 Materials and methods

sponding columns of any two compared motifs. For example, a test motif which contains only a small number of strongly conserved residues cannot be expected to have a good scanning performance in identifying all spiked motifs, because the motif tolerates too much sequence variation. Therefore judging the performance of a motif discovery tool based on only such sensitivity/specificity measures is inadequate, since a motif tool should find a weak motif from a set of spiked data, if the original motif is a weak one, too. The sensitivity/specificity of this type of less conserved motifs would be relatively low, and not reflect or reward a program's ability to have discovered such a difficult motif. Therefore, I report MCC of the original test motifs primarily as a measure indicating how difficult a motif is to recover by a motif discovery program, and I report Cartesian motif distances with the purpose of indicating how good the program is in that task. For instance, even an MCC value of 0.65 would still be good for a motif found by a program, if the corresponding real test motif did not have a much better MCC.

To generate test motifs for the program's assessment, I used conserved blocks of several ClustalW multiple alignments of sufficiently large number of Swiss-Prot (Bairoch & Apweiler, 1996) proteins. These proteins feature arbitrarily chosen Prosite (Hulo *et al.*, 2006), or PFAM domain entries. Segments from these domains' alignments were converted into PWMs to obtain 3 sets of 7 test motifs of varying lengths between 3 and 9. The 21 test motifs used in the evaluations are available for download at the NestedMICA home page.

As a dataset to carry out the spiking tests on, I used 438 whole-length cytoplasmic protein sequences obtained from the redundancy-reduced non-plants version of the TargetP (Emanuelsson *et al.*, 2000) subcellular localisation dataset.

Having an average sequence length of 582, this dataset does not include any homologous proteins, after a filtering process performed as suggested by [Hobohm *et al.* \(1992\)](#). Both NestedMICA and MEME were run with the default options. Note that, NestedMICA’s default parameters differ from those used in DNA motif finding. Both NestedMICA and MEME require a target motif length interval, and no matter what the actual spiked motif’s length was, for all of our spiking tests this was set to be between 3 and 15.

The background model used in the spiking tests was trained from the same cytoplasmic sequence dataset. The similar background likelihood analysis that was performed on another set ([Figure 2.1](#)) suggested that there would be no significant gain in likelihood when using a model with more than 4 mosaic classes for this particular small dataset. Therefore, a first order background model containing 4 mosaic classes was used in the tests.

Finally, for the evaluation of the program’s assessment in subcellular localisation motif recovery, which was performed using sequences of different lengths, I used the ER dataset of a multi-class protein subcellular localisation predictor, MultiLoc ([Höglund *et al.*, 2006](#)). This dataset contains 198 homology-reduced, eukaryotic ER proteins.

2.3 Results and discussions

2.3.1 Protein sequence background model

The first step in using NestedMICA is the generation of a background model to represent the uninteresting parts of sequences that do not contain motifs of interest (see methods). From a series of tests I concluded that different sets of protein

sequences vary in complexity and composition too much to develop a generic background model. Most of the time, training a dedicated background model for each protein dataset is the best way to maximise performance and sensitivity. Prior to motif finding, sequence likelihood analysis must be performed to test a variety of background models and select the optimal one. Figure 2.1 shows one such likelihood curve performed on a set of cytoplasmic proteins. Generally, if there is sufficient data to perform a proper training, using order-1 background models proved to be better than order-0 models for proteins. As far as the number of mosaic classes is concerned, a class number should be picked that falls on the corresponding likelihood curve before it starts to saturate or drop, regardless of whether it increases at a later stage.

2.3.2 Performance vs. motif abundance

I used 3 different motif sets each containing 7 motifs of lengths ranging from 3 to 9 amino acids. Instances of each of the motifs depicted in Figures 2.2, 2.3 and 2.4 (for motif sets 1, 2 and 3, respectively) were separately spiked into the cytoplasmic dataset (see Section 2.2.5). The 21 motifs were inserted into the sequences at different frequencies (10, 20 and 30%), allowing us to test motif discovery software under different conditions of motif abundance. Generally, performance for both NestedMICA and MEME increased with increasing abundance rate of the inserted motif.

Each of these three figures shows a set of tests performed at different motif abundance rates with the original test motifs, along with the corresponding motifs found by both NestedMICA and MEME. For each motif reported by NestedMICA

2.3 Results and discussions




































Original motif	Abundance	MCC for original	NestedMICA	Distance & MCC for NestedMICA	MEME	Distance & MCC for MEME
	10	0.753		0.57 0.830		
	20			0.34 0.830		
	30			0.33 0.830		
	10	0.856		2.70 0.153		
	20			3.72 0.015		
	30			0.72 0.537		
	10	0.749		1.58 0.499		
	20			0.50 0.699		
	30			0.55 0.723		
	10	0.815		5.67 0.011		
	20			0.71 0.648		
	30			0.70 0.653		
	10	0.918		5.10 0.015		
	20			0.78 0.816		
	30			0.68 0.795		
	10	0.993		0.80 0.926		
	20			0.52 0.935		
	30			0.52 0.935		
	10	0.990		5.21 0.118		
	20			1.00 0.784		
	30			0.93 0.795		

Figure 2.2: Motifs recovered by NestedMICA and MEME in the single-motif spiking tests, for motif set1. Motifs in this set were obtained from several Pfam domain entries. For each original test motif used in the motif spiking tests, the 3 tested abundance rates are shown in the next column. For motifs recovered by NestedMICA (fourth column) and MEME (sixth column) the Cartesian distance to the original test motif and the MCC value obtained when the motif is used for sequence scanning are shown. For comparison purposes, the MCC values of the original test motifs are shown as well. In NestedMICA protein sequence logos, hydrophobic residues are represented in orange, polar and hydrophilic ones in green, acidic ones in pink, and finally basic amino acids are depicted in blue.

2.3 Results and discussions

Original motif	Abundance	MCC for original	NestedMICA	Distance & MCC for NestedMICA	MEME	Distance & MCC for MEME
GPF	10	0.850		0.84 0.726		
	20			0.35 0.726		
	30			0.26 0.850		
KYGV	10	0.822		4.50 0.018		
	20			0.69 0.693		
	30			0.32 0.693		
ATCP	10	0.921		0.76 0.931		
	20			0.50 0.936		
	30			0.29 0.884		
W ₁ YKDQ	10	0.884		0.81 0.911		
	20			0.51 0.858		
	30			0.47 0.886		
Q ₁ DR ₁ DK	10	0.943		1.03 0.939		
	20			0.64 0.939		
	30			0.45 0.948		
YLR ₁ LP ₁ P	10	0.830		5.45 0.012		
	20			0.71 0.810		
	30			0.61 0.812		
REGLYR ₁ SG	10	0.941		6.85 0.016		
	20			0.99 0.927		
	30			0.67 0.932		

Figure 2.3: Motifs recovered by NestedMICA and MEME in the single-motif spiking tests, for motif set2. Motifs in this set were obtained from several Prosite domain entries. For each original test motif used in the motif spiking tests, the 3 tested abundance rates are shown in the next column. For motifs recovered by NestedMICA (fourth column) and MEME (sixth column) the Cartesian distance to the original test motif and the MCC value obtained when the motif is used for sequence scanning are shown. For comparison purposes, the MCC values of the original test motifs are shown as well.

2.3 Results and discussions

Original motif	Abundance	MCC for original	NestedMICA	Distance & MCC for NestedMICA	MEME	Distance & MCC for MEME
	10	0.753		1.11 0.539		Distance & MCC for MEME
	20			0.14 0.753		
	30			0.10 0.753		
	10	0.856		3.22 0.037		
	20			0.53 0.758		
	30			0.37 0.730		
	10	0.749		0.63 0.659		
	20			0.58 0.673		
	30			0.34 0.708		
	10	0.815		1.11 0.750		
	20			0.58 0.780		
	30			0.55 0.761		
	10	0.918		0.99 0.857		
	20			0.77 0.873		
	30			0.43 0.890		
	10	0.993		0.59 0.990		
	20			0.34 0.990		
	30			0.23 0.990		
	10	0.990		1.10 0.988		
	20			0.63 0.993		
	30			0.56 0.990		

Figure 2.4: Motifs recovered by NestedMICA and MEME in the single-motif spiking tests, for motif set3. Motifs in this set were obtained from several Pfam domain entries. For each original test motif used in the motif spiking tests, the 3 tested abundance rates are shown in the next column. For motifs recovered by NestedMICA (fourth column) and MEME (sixth column) the Cartesian distance to the original test motif and the MCC value obtained when the motif is used for sequence scanning are shown. For comparison purposes, the MCC values of the original test motifs are shown as well.

2.3 Results and discussions

and MEME, its Cartesian distance from the corresponding original motif is given. As Tables 2.1 and 2.2 summarise, low abundance motifs and short motifs were more difficult to recover for MEME, even if they had a high information content. For example, out of the maximum 4.32 bits per position, the average information content per position was 3.96 bits (91.5%) for motif of length 3 in set 2, while it was 3.68 bits (85.2%) for motif of length 4 in the same motif set (Figure 2.3). Both could not be recovered by MEME at the tested 10, 20 and 30% abundance rates. The motif of length 3, for example, could only be recovered correctly by MEME when it was present in at least 80% of the sequences (data not shown). In contrast, the same motif was recovered by NestedMICA when present in only 10% of the sequences. NestedMICA did not miss any of the 21 motifs when they were present at 30% abundance. It also correctly recovered 95.2% and 61.9% of them when the motif abundance rate was 20%, and 10%, respectively (Table 2.2).

Spiked in(%)	Set 1		Set 2		Set 3	
	NestedMICA	MEME	NestedMICA	MEME	NestedMICA	MEME
10	3	0	4	0	6	2
20	6	1	7	4	7	3
30	7	3	7	5	7	4

Table 2.1: **Motif recovery performance for NestedMICA and MEME for individual test sets.** Numbers shown correspond to the correctly recovered number of motifs for each test set, each of which contains 7 motifs, for the single-motif spiking tests. Motifs recovered for set 1, 2 and 3 can be seen on Figures 2.2, 2.3, and 2.4, respectively. A motif is considered as correctly recovered if the average Cartesian distance per residue position between the recovered motif and the original motif that was spiked is < 0.3 (see Section 2.2.5).

In addition to Cartesian motif distances, measuring the similarity between the recovered motif and the original, the performance of the motifs in finding motif

2.3 Results and discussions

instances when scanning test sequences is indicated by Matthew’s Correlation Coefficient (MCC) (Matthews, 1975) values (Figures 2.2, 2.3 and 2.4). The MCC is a single measure that captures performance over a range of sensitivity and specificity values (see methods). Raw sensitivity and specificity values are given Tables 2.3, 2.4 and 2.5 for all three motifs sets. These measures have been used to evaluate the scanning performances of the original and reported motifs, by testing spiked datasets (independent of the spiked datasets used for training) where each sequence contains an instance of a particular motif. I provide the MCC values for the original test motifs, too, for better interpretation of the MCC values given with the motifs reported by both programs. Having relatively lower sensitivity / specificity values, and hence a lower MCC, does not necessarily mean that a program is not doing well in finding a certain motif, but in certain cases it can indicate that the target motif is a weak one and therefore more difficult to recover. MCC values for the original motifs were calculated in a similar way to the others, i.e., by spiking every sequence in the background test dataset with the generated instances of a particular motif, and then scanning the spiked dataset with the original motif to see how many motif hits would be found using a range of score

Motif abundance(%)	Total correct (%)	
	NestedMICA	MEME
10	61.9	9.5
20	95.2	38.0
30	100.0	57.1

Table 2.2: **Total motif recovery performance summary for NestedMICA and MEME.** Percentages of correctly recovered motifs are given for the 3 motif abundance rates tested, considering all 21 test motifs from three of the sets.

thresholds (see methods).

Length	Abundance	NestedMICA		MEME	
		SN	SP	SN	SP
3	10	0.988	0.855	0.995	0.501
3	20	0.988	0.855	0.995	0.501
3	30	0.988	0.855	0.995	0.501
4	10	0.811	0.545	0.197	0.506
4	20	0.995	0.501	0.197	0.506
4	30	0.847	0.728	0.197	0.506
5	10	0.487	0.914	0.592	0.507
5	20	0.753	0.921	0.592	0.507
5	30	0.782	0.921	0.592	0.507
6	10	0.950	0.501	0.978	0.501
6	20	0.849	0.808	0.978	0.501
6	30	0.703	0.913	0.978	0.501
7	10	0.995	0.501	0.995	0.501
7	20	0.890	0.923	0.995	0.501
7	30	0.823	0.958	0.818	0.950
8	10	0.957	0.968	0.856	0.507
8	20	0.959	0.976	0.959	0.976
8	30	0.971	0.964	0.964	0.969
9	10	0.974	0.514	0.990	0.502
9	20	0.835	0.938	0.995	0.501
9	30	0.875	0.915	0.851	0.939

Table 2.3: **Sensitivity (SN) and specificity (SP) values for motifs of Set 1, reported by NestedMICA and MEME in the single-motif spiking tests.** Length refers to number of residue positions in motifs.

2.3.3 Performance with multiple motifs

Individual protein sequences may contain multiple different motif of interest. For example, proteins targeted into the endoplasmic reticulum (ER) by an N-terminal Signal Peptide (SP) sequence are maintained in the ER if they have also a [KH]DEL retention signal on their C-terminus. After determining the ability of

Length	Abundance	NestedMICA		MEME	
		SN	SP	SN	SP
3	10	0.770	0.933	0.990	0.502
3	20	0.770	0.933	0.990	0.502
3	30	0.950	0.904	0.995	0.501
4	10	0.930	0.503	0.942	0.504
4	20	0.664	0.986	0.988	0.501
4	30	0.842	0.850	0.988	0.501
5	10	0.978	0.953	0.995	0.501
5	20	0.954	0.980	0.995	0.501
5	30	0.974	0.914	0.986	0.895
6	10	0.921	0.987	0.935	0.502
6	20	0.866	0.984	0.918	0.958
6	30	0.914	0.969	0.871	0.976
7	10	0.947	0.990	0.866	0.503
7	20	0.942	0.995	0.952	0.978
7	30	0.962	0.985	0.957	0.964
8	10	0.959	0.501	0.974	0.504
8	20	0.873	0.931	0.861	0.940
8	30	0.873	0.933	0.851	0.947
9	10	0.995	0.501	0.998	0.501
9	20	0.940	0.985	0.935	0.975
9	30	0.938	0.992	0.957	0.980

Table 2.4: Sensitivity (SN) and specificity (SP) values for motifs of Set 2, reported by NestedMICA and MEME in the single-motif spiking tests.

2.3 Results and discussions

Length	Abundance	NestedMICA		MEME	
		SN	SP	SN	SP
3	10	0.559	0.903	0.197	0.506
3	20	0.875	0.877	0.197	0.506
3	30	0.875	0.877	0.197	0.506
4	10	0.921	0.506	0.993	0.501
4	20	0.839	0.909	0.993	0.501
4	30	0.775	0.934	0.993	0.501
5	10	0.731	0.897	0.854	0.506
5	20	0.782	0.874	0.854	0.506
5	30	0.837	0.866	0.854	0.506
6	10	0.839	0.902	0.995	0.501
6	20	0.863	0.911	0.995	0.501
6	30	0.794	0.948	0.856	0.932
7	10	0.906	0.947	0.139	0.532
7	20	0.882	0.984	0.926	0.977
7	30	0.902	0.984	0.928	0.968
8	10	0.995	0.995	0.995	0.995
8	20	0.993	0.998	0.995	0.998
8	30	0.993	0.998	0.990	1.000
9	10	0.993	0.995	0.986	1.000
9	20	0.995	0.998	0.995	0.998
9	30	0.995	0.995	0.995	0.995

Table 2.5: **Sensitivity (SN) and specificity (SP) values for motifs of Set 3, reported by NestedMICA and MEME in the single-motif spiking tests.**

Name	Motif	MCC
m4		0.82
m7		0.94
m10		0.97

Figure 2.5: **Inserting more than one different motif into the sequences.** Original motifs used in multiple motif test are shown. These were inserted into the test sequences, at 40 and 20% total motif abundance rates. Resulting spiked sequences contain either zero, one or multiple different instances of the shown motifs, while sequences were not allowed to contain multiple instances of the same motif. The MCC values of these original motifs are given for comparison with the recovered motifs' MCCs. Results for recovered motifs are presented in Tables 2.6 and 2.7.

both NestedMICA and MEME to find single motifs, I assessed the two programs' ability to recover multiple motifs from a single dataset.

I used 3 test motifs of length 4, 7 and 10 aa, in the multiple motif spiking tests (Figure 2.5). Multiple motifs were spiked in such a way as to ensure an unbiased distribution. For example, in the first multiple motif spiking test, corresponding to a 40% abundance rate for each motif, it was ensured that 24% of the sequences were spiked with only motif of length 7, 24% only with motif of length 10 and 16% with both motifs. This corresponds to the distribution of motifs that would be expected by chance. The test was repeated by halving the total abundance rate for each motif.

In a similar way, two other pair combinations of the motifs were tested, and

2.3 Results and discussions

finally, three motifs were spiked at the same time. When the abundance rate for each spiked motif in the triple motif test was 40%, it was ensured that three different groups of sequences, each corresponding to 14.4% of the total, contained either motif of length 4, or 7 or 10; three different groups, each corresponding to 9.6% of the total contained two motif instances simultaneously (i.e. one group had both motifs of length 4 and 7, another had both 7 and 10, and finally another had both 4 and 10) and one group corresponding 6.4% contained all three motifs.

Tables 2.6 and 2.7 summarise the performances of NestedMICA and MEME, respectively, for the multiple motif finding tasks performed under different conditions. It shows the Cartesian distances and MCC values of the reported motifs (The corresponding sensitivity and specificity values are given in Table 2.8 for NestedMICA and Table 2.9 for MEME). In general, both NestedMICA and MEME performed well, except MEME had a tendency not to recover shorter motifs and instead report PWMs of maximum allowed length which did not correspond to any of the spiked motifs.

Motifs	Abundance	Distances	MCCs
m4 + m7	40	0.23, 0.45	0.74, 0.93
	20	0.54, 0.62	0.71, 0.93
m4 + m10	40	0.44, 0.75	0.81, 0.95
	20	0.34, 0.73	0.75, 0.96
m7 + m10	40	0.47, 1.11	0.95, 0.96
	20	0.71, 0.75	0.93, 0.95
m4 + m7 + m10	40	0.42, 1.01, 1.00	0.75, 0.95, 0.97
	20	0.64, 0.54, 0.57	0.71, 0.95, 0.97

Table 2.6: **Performance summary for NestedMICA in the multiple motif spiking tests.** The “distances” columns refer to the Cartesian distances between the reported motifs and the original ones which are shown in Figure 2.5. Motif names indicate length. In addition to Cartesian distances, MCC values are given for motifs recovered by NestedMICA.

2.3 Results and discussions

Motifs	Abundance	Distances	MCCs
m4 + m7	40	11.73, 0.53	0.02, 0.92
	20	11.73, 0.56	0.02, 0.94
m4 + m10	40	11.73, 0.46	0.02, 0.96
	20	11.73, 0.75	0.02, 0.96
m7 + m10	40	0.38, 0.45	0.94, 0.95
	20	0.70, 0.62	0.92, 0.95
m4 + m7 + m10	40	11.73, 0.44, 0.42	0.02, 0.93, 0.96
	20	11.73, 0.76, 0.82	0.02, 0.93, 0.95

Table 2.7: **Performance summary for MEME in the multiple motif spiking tests.** The “distances” columns refer to the Cartesian distances between the reported motifs and the original ones which are shown in Figure 2.5. Motif names indicate length.

Motifs	Abundance (%)	NestedMICA	
		SN	SP
m4 + m7	40	0.892, 0.949	0.855, 0.980
	20	0.685, 0.947	0.986, 0.980
m4 + m10	40	0.973, 0.964	0.856, 0.985
	20	0.745, 0.978	0.974, 0.980
m7 + m10	40	0.968, 0.976	0.982, 0.987
	20	0.932, 0.971	0.994, 0.983
m4 + m7 + m10	40	0.978, 0.968, 0.976	0.798, 0.985, 0.990
	20	0.685, 0.964, 0.978	0.986, 0.980, 0.987

Table 2.8: **Sensitivity (SN) and specificity (SP) values for motifs reported by NestedMICA in the multiple-motif spiking tests.** Motif names (m4, m7 etc.) refer to length and are shown in Figure 2.5. SN and SP values are given for each of the motifs involved in a multiple motif spiking test, and are separated by commas.

2.3.4 Performance vs. protein length

Having performed the motif spiking tests, in order to evaluate the two programs in a more natural situation, I observed the effects of varying sequence length on motif finding in multiple protein sets expected to contain C-terminal motifs. To this

2.3 Results and discussions

end, I used 198 non-redundant ER proteins (see Methods), a high proportion of which would be expected to contain the C-terminal ER retention signal mentioned above. I created three datasets containing sequence chunks of 60, 80 and 100 amino acid letters, respectively, taken from the C-terminal regions of these ER proteins.

Figure 2.6 depicts the motifs recovered from these three datasets by both programs. While MEME could not find the [KH]DEL motif at the tested sequence lengths of 80 and 100 amino acids, NestedMICA performed well, even when 100 amino acid long chunks were used. Apart from not looking similar at all to the KDEL motif, there was no consistency between the motifs reported by MEME when using the 80 and 100aa long sequences. Both programs were run with default protein parameters with a target motif length set to between 3 and 15 amino acids.

To investigate whether NestedMICA would still find the motif when there are

Motifs	Abundance (%)	MEME	
		SN	SP
m4 + m7	40	0.942, 0.947	0.503, 0.975
	20	0.942, 0.959	0.503, 0.982
m4 + m10	40	0.942, 0.980	0.503, 0.985
	20	0.942, 0.988	0.503, 0.978
m7 + m10	40	0.954, 0.980	0.982, 0.976
	20	0.949, 0.978	0.975, 0.973
m4 + m7 + m10	40	0.942, 0.952, 0.978	0.503, 0.980, 0.985
	20	0.942, 0.954, 0.976	0.503, 0.975, 0.978

Table 2.9: **Sensitivity (SN) and specificity (SP) values for motifs reported by MEME in the multiple-motif spiking tests.** Motif names (m4, m7 etc.) refer to length and are shown in Figure 2.5. SN and SP values are given for each of the motifs involved in a multiple motif spiking test, and are separated by commas.



Figure 2.6: **Motif recovery performance against sequence length.** The figure shows recovered motifs using NestedMICA and MEME. “Length” refers to how many amino acid letters from the right-most (C-terminal) part of sequences were used in each dataset created. The 4 amino acid long ER retention signal was recovered successfully by NestedMICA while MEME reported motifs of the maximum allowed length (given by the user) when the sequences were longer than 80 residues.

more than 100 residues per sequence, I tested it using 120 residue long C-terminal regions. The ER retention motif was found only when NestedMICA was asked to find two motifs. Investigating the other reported motif, I found that it was a thioredoxin family active site motif (Prosite id: PDOC00172) that is usually found in ER proteins. MEME was also tested when forced to find two motifs from the dataset containing the 80 amino acid long sequences. However, in addition to the motifs shown in Figure 2.6, it reported a 15 residue long motif which I could not locate in domain databases. Scanning this motif against the sequences, I noticed that it exists in 8 of the 198 proteins in the dataset.

2.3.5 “Null test” and significance of motifs

For motif discovery assessment purposes, spiking motifs into a dataset of sequences that already contained strong motifs would be undesirable, as the method

in question might report some of these intrinsic motifs instead of the artificially implanted ones. On the other hand, evaluating a motif discovery tool using a dataset of randomly generated sequences would be unfair, too, as this would be relatively easy for the program to recover a test motif.

Given that even sequences having a low sequence identity can in theory share some common sequential features, it is important to ensure that an unbiased set of sequences is used in the tests. For this reason I used non-homologous cytoplasmic sequences from the TargetP subcellular localisation dataset for these tests. This dataset had been already filtered by the TargetP developers using a homology reduction algorithm (Hobohm *et al.*, 1992) that ensures no homologous sequences exist in the set (Emanuelsson *et al.*, 2000), before I filtered it again to further reduce the maximum sequence identity between any of the sequences.

I ran both NestedMICA and MEME on this dataset, before it was spiked by any test motifs, using different minimum target motif lengths for each program tested. This “null test” was performed to confirm that the dataset I used in performing motif spiking tests is a reasonably suitable one. This negative control test also gives an idea about how well the trained background model represented the sequences.

For this purpose, NestedMICA was run with the default parameters optimized for protein sequences (for more details on the parameters, please see the program manual). In this test, the minimum target length was initially set to 2, then 3, and finally 4, while the maximum length was always kept as 15, as in the motif spiking tests. Motifs generated by NestedMICA from these runs were weak (Figure 2.7), having average information bit scores per position not exceeding

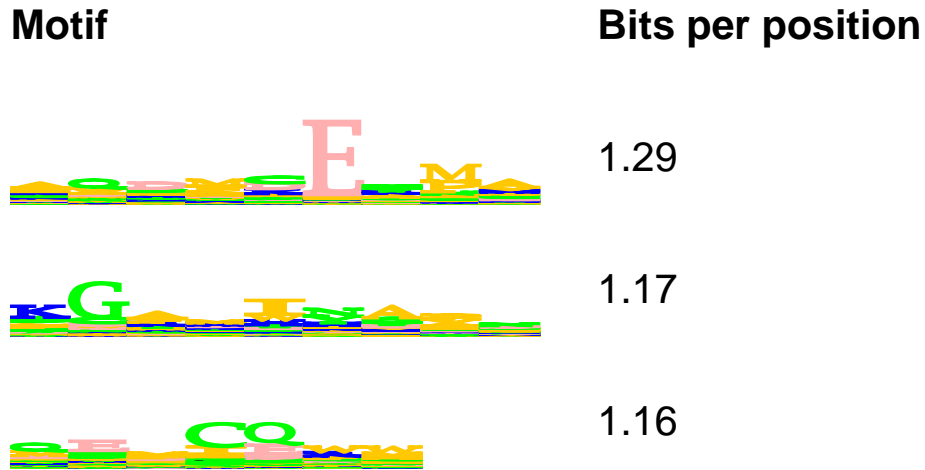


Figure 2.7: NestedMICA’s “null motifs”. When the minimum length parameter was set to 2, 3, and 4, NestedMICA generated almost flat motifs with few conserved positions, when no motif was inserted into the cytoplasmic test dataset. Bits per position is the averaged out value for the total information content of a motif, where it could be a maximum of 4.32 bits per position.

1.3 out of the possible 4.32 bits per position, which corresponds to roughly less than one third of the maximum height in a sequence logo. This indicates that NestedMICA does not generally report false positive motifs, and that the chosen background model parameters are good enough to represent the test set. As we have seen above, NestedMICA is sensitive enough to report even scarce motifs of length 3 when present in only 10% of the sequences, as the examples in Figures 2.2 - 2.4 indicate. Therefore, the fact that NestedMICA only reports weak “null test” motifs increases our confidence that the cytoplasmic sequence set that I use to assess motif discovery performance is not likely to contain significant motifs that a motif finder would prefer to report over any of our spiked motifs.

MEME, on the other hand, generally tended to report high-information containing motifs of the maximum allowed length, corresponding to about 46 bits in

total, and above 3 bits per residue position. To minimize any remaining common patterns in the sequence set, I further reduced the maximum sequence identity within the set to 30%. Furthermore, all sequence regions matching a Prosite pattern were removed, based on hits reported by an annotated motif search tool PPSearch (Quevillon *et al.*, 2005). However, even with this extra filtered dataset, MEME still reported strong and long motifs similar to the 15 amino acid long ones in Figures 2.2 - 2.4.

When the user-specified number of target motifs exceeds the number of actual motifs, NestedMICA has been observed to generate motifs that look like the null motif of that particular dataset (data not shown). Similarly, MEME produced the same type of long motifs it found in the null tests when it failed to find an inserted motif in the spiking tests.

2.3.6 Testing non-*ab initio* motif finders

As mentioned in the introduction section of this chapter, there are protein discovery tools which are not in the *ab initio* motif discovery category because they either might be using database look-ups, or homology search etc. One such program is Dilimot (Neduva & Russell, 2006). However, in addition to searching databases including PFAM (Bateman *et al.*, 2004) and SMART (Schultz *et al.*, 1998), it also utilises an *ab initio* tool, called TEIRESIAS (Rigoutsos & Floratos, 1998), which finds and lists frequently occurring patterns that could even contain gaps. Motifs are not reported as PWMs by this program. I normally compared NestedMICA with another probabilistic, *ab initio* method, MEME, which outputs motifs as PWMs, too. In this section, I provide an example to show whether

tools generating regular expressions for describing discovered patterns could be used successfully as the other probabilistic methods.

The Dilimot web server was provided with one dataset of protein sequences, 30% of which were spiked with motif of length 3 from the motif set 1 that I used to assess other programs (see Figure 2.2). The dataset contained 409 redundancy-reduced cytoplasmic sequences taken from the targetP (Emanuelsson *et al.*, 2000) subcellular localisation training set. After running a couple of days, the Dilimot program produced a table of discovered patterns in the form of regular expressions (Figure 2.8), however, none of the reported motifs were similar to the artificially spiked motif.

One disadvantageous aspect of such programs is that they are not based on probabilistic background models, which makes it very difficult for them to recover less abundant and short functional motifs, if not merely impossible. Because of this reason, they may report frequently repeating sequence regions instead, or regions that could be related to compositional features, unless they use motif databases having an entry for that particular motif.

2.4 Conclusions

I have added support for protein motif discovery in NestedMICA. It reports protein motifs in the form of PWMs. It has been optimized for better protein motif discovery under stringent conditions, and automatic motif length adjustment. In summary, our performance assessment tests show that NestedMICA performs very well when finding single and multiple motifs even at low motif abundance rates and different motif lengths, thus proving itself to be a robust and sensitive

2.4 Conclusions

Input parameters: Job id QLMD_22970. Filters: GlobPlot No, Redundancy Yes, Pfam Yes, SMART Yes.
 Motif parms: fixed pos. 3; maximum len 8; min no. proteins w. motif (ie col. 3) 3.
 Species contributing to Scons: F.rubripes M.musculus R.norvegicus G.gallus H.sapiens D.melanogaster A.gambiae
 C.elegans A.thaliana S.pombe K.lactis C.albicanis S.cereviciae D.hansenii K.waltii S.cereviciae C.glabrata

Display criteria: Max Scons 1.00e-05; Min. seq. w. motif (col. 3): 4; Max motifs shown 10

Motif	S _{cons} (help)	Proteins with motif (in unfiltered regions) (help)	Proteins total (non-redundant) (help)	P (binomial test) (help)
<input type="checkbox"/> GxExF	0.00e+00	24	409	0.00e+00
<input type="checkbox"/> NxxxCxK	0.00e+00	5	409	0.00e+00
<input type="checkbox"/> GGGGGGxG	0.00e+00	4	409	0.00e+00
<input type="checkbox"/> LAxxxAxR	0.00e+00	4	409	0.00e+00
<input type="checkbox"/> DxxSSSS	0.00e+00	5	409	0.00e+00
<input type="checkbox"/> DxxxxEKQ	0.00e+00	4	409	0.00e+00
<input type="checkbox"/> HxALxxN	0.00e+00	5	409	0.00e+00
<input type="checkbox"/> FxxxxKxG	0.00e+00	27	409	0.00e+00
<input type="checkbox"/> PPPxPPxP	0.00e+00	6	409	0.00e+00
<input type="checkbox"/> SxVxxLxS	0.00e+00	9	409	0.00e+00
<input type="button" value="Apply new selection"/>	Display graphically all motifs in this table (help)			

Figure 2.8: A snapshot showing the regular expressions reported by the Dilimot web service. Dilimot was run with the default options. It was allowed both to use the *ab initio* program, TEIRESIAS, and to consult other public protein pattern databases.

2.5 Availability and requirements of NestedMICA

protein motif finder. Judging from the calculated sensitivity, specificity and MCC values, there was no clear difference regarding the quality of motifs correctly recovered by NestedMICA or MEME. However, when it comes to the number of correctly recovered motifs, NestedMICA significantly outperformed MEME in our protein motif finding tasks including finding low abundant motifs, finding short motifs, and finally discovering motifs from amino acid sequences of different lengths.

In addition to assessing its ability in finding true positive motifs, as shown in the results section, by running it on a non-redundant dataset where no test motif was inserted, I have shown that NestedMICA does not tend to report high-information content motifs when there is no meaningful motif contained in the dataset, i.e. that it tends not to report strong false negatives.

Considering that some protein signals such as subcellular localisation motifs could be as short as 3 amino acids, this new protein motif finder is a promising tool in functional sequence annotation.

2.5 Availability and requirements of NestedMICA

- **Project Name:** NestedMICA
- **Project home page:** <http://www.sanger.ac.uk/Software/analysis/NestedMICA/>
- **Operating systems:** Platform independent
- **Programming language:** Java
- **Other requirements:** Biojava1.4, WoodStox, StAX-compliant XML parser

2.5 Availability and requirements of NestedMICA

(all included within the NestedMICA package), ANT 1.7.0 (<http://ant.apache.org>)
to compile the project

- **License:** LGPL
- **Any restrictions to use by non-academics:** None