Chapter 4

Metamotifs in motif classification

Metamotifs are shown in the previous chapter to significantly improve the sensitivity to infer motifs from sequence, when applied as a Bayesian PWM prior in the NestedMICA algorithm. Here I will show that metamotifs can also be applied to form functional predictions for motifs. Metamotifs are applied to a motif classification problem where features extracted from regulatory motifs (PWMs) are used to predict the family of protein DNA binding domains which is likely to interact with them. I will refer to this problem as 'motif family classification'. The features I used in my motif family classifier are metamotif densities, and I therefore call the method **metamatti**, for **metam**otif based **a**utomated **t**ranscription factor **t**ype **i**nference.

4.1 Previous work on motif family classification

Motif family classification is not a new idea. In particular, the following three studies provided an inspiration for the work described here:

- The hidden Markov Dirichlet-multinomial based MotifPrototyper framework (Xing and Karp, 2004), which is also used to provide the PWM column-specific Bayesian prior function discussed in the previous chapter (Section 3.1). The MotifPrototyper based motif classification is presented as a cross-validation based exercise where motifs from the TRANSFAC database are labelled with their superclass (one of basic, zinc coordinated,

helix-turn-helix, or β-scaffold domains, see Section 1.4.2 for a further discussion on structural taxonomies of TFs). The ability to classify motifs on the level of their superclass is discussed by Xing and Karp (2004) mostly as an interesting side-product of co-evolution of transcription factors and their binding sites, and the authors do not make available the motif classifier for other researchers to use.

- The sparse multinomial logistic regression (SMLR) based motif classifier by Narlikar and Hartemink (2006). Similarly as above, the emphasis of this work is not in constructing a publicly available motif family classification tool for the research community, but to present the classification problem as a side-product of the evolutionary pressures acting of TFs and their binding sites. The paper also acts as a biological application to a novel sparse, probabilistic supervised machine learning method developed by the authors (SMLR). The classification is done, as in the case of MotifPrototyper, to motifs in the TRANSFAC database (its six largest classes Cys_2His_2 and Cys_4 zinc fingers, homeodomains, forkhead domains, basic helix-loop-helices and basic zipper domains), but the classifier labels the motifs with their TRANSFAC class (not superclass, as done by MotifPrototyper). Notably, the same authors also published a separate paper (Narlikar et al., 2006) where they present an informative PWM prior enabled motif inference algorithm which also labels the discovered motifs with their domain family. This paper is discussed in the context of motif priors in Chapter 3.

- Sandelin and Wasserman (2004) are the earliest at suggesting a computational motif family labelling method, in the same familial binding profile paper which was discussed in the previous motif prior chapter. It is however limited to a small number of metazoan TFs (63 in total) which are closely similar in the clustering chosen by the authors (bZIP motifs for instance are subdivided to three subgroups). Due to the limited scope of this classification study, and the biased choice of the motifs in this study, I decided not to assess my method against it (similar choice was also made by Narlikar and Hartemink (2006)).

In contrast to the previous studies, my goal in this work has been to both rigorously test my method in context of the earlier work where applicable, and to also present a tool for motif family classification that other researchers can use in the comparative study of regulatory motifs. Indeed, **metamatti** can be distributed as an R package (Section 4.3.4.1), and as a remotely available motif classification web server (Section 4.3.5).

In this chapter I firstly introduce the **metamatti** classifier, and compare its performance to two of the methods noted above: MotifPrototyper (Xing and Karp, 2004) and SMLR (Narlikar and Hartemink, 2006). I also validate the classification method's performance with two independent, experimentally validated homeodomain datasets, and give a brief introduction to the usage of the classification tool. In the next chapter I then apply the method to a series of computationally predicted motifs, to showcase **metamatti**'s ability to predict the class of *de novo* predicted motifs from a genome scale motif inference study. In addition to assigning clues of function to large sets of de novo motifs, I believe that family classification of motifs could for instance become a useful diagnostic method when working with TFBS motifs predicted from genomic ChIP-chip or ChIP-seq data; with it, one could test how closely motifs predicted from the DNA fragments bound by a TF of interest match the expected familial pattern of the DNA binding domain under study. This can be helpful in identifying the relevant motif from potentially many that are over-represented in DNA fragments bound in a ChIP assay. This idea has been explored by MacIsaac et al. (2006) with a familial binding profile based method.

4.2 Materials & Method

The principle of my motif classifier is to compute the density function (Equation 2.5) of a large dictionary of familial metamotifs along the length of training set motifs, effectively "scanning" weight matrices with metamotifs. The optimal (maximum) and average metamotif densities of each metamotif with the motif are then included as features in a random forest classifier that tries to infer the TRANSFAC superfamily (Figure 4.1) or TRANSFAC family (Figure 4.2) of the motifs. Random forest classification was chosen as the machine learning frame-

work, most importantly because it generalises naturally to multi-class problems and provides reliable error estimates as part of model training (Breiman, 2001b). The framework also controls the sparsity of the feature set used for classification (see Section 1.3.4 for an introduction to random forests).

4.2.1 Training data

All motif families with at least 10 representatives were retrieved from the TRANSFAC 12.2 database (Matys et al., 2006), totalling 623 motifs of 13 domain families (see Section 1.4.2.1 for more information about the TRANSFAC database). For the motif domain superfamily classifier comparison made with MotifPrototyper Xing and Karp (2004) (Figure 4.1), the set of motifs was reduced further to include only motifs annotated in TRANSFAC with the four superfamilies classified in (Xing and Karp, 2004). For the motif TRANSFAC class prediction comparison with SMLR (Figure 4.2), only motifs of the same six major classes classified with SMLR in Narlikar and Hartemink (2006) were included in our training set. The feature set is discussed in Section 4.2.2.

The **metamatti** motif type classifier training and cross-validation were implemented in the Ruby and R (Team, 2007) programming languages. Random forest classification was done using the package `randomForest` (Liaw and Wiener, 2002). Pseudocounts of 0.01 were added to all training set metamotifs, and the *mtry* parameter of the random forest classifier training was optimised by testing $0.1 \times \sqrt{p}, 0.2 \times \sqrt{p} \ldots, 2.0 \times \sqrt{p}$ with intervals of 0.1, where p is the number of features in the classifier (the default value for *mtry* is $\sqrt{p}$). The *ntree* parameter that controls the number of trees to grow was set at 5000.

4.2.2 The classifier feature set

Most features in **metamatti** are metamotif probability density scores (Table 4.1). To compute the metamotif density features for the classifier, we chose to first divide the motifs into sets by complete linkage hierarchical clustering (Johnson, 1967) with the SSD metric described in Down et al. (2007) and cutting the clusters at a lenient clustering cutoff of 6.0. This resulted in 54 motif clusters. Three metamotifs were trained from each motif cluster with `nmmetainfer`, resulting in

195 metamotifs to be used in the motif classifier (examples seen in Figure 2.10). Metamotif length was constrained between 6 and 15 columns, and the expected usage fraction was set at 0.5.

Feature type	**Description**
Maximum metamotif hit scores with all of the familial metamotifs	Motifs were scanned with all input metamotifs, and the optimal score was chosen.
Per-column average entropy	Average Shannon entropy of columns.
MLE Dirichlet parameters	A maximum likelihood Dirichlet distribution is estimated as described in Minka (2003), and the parameters of this distribution are used as features(α_A,α_G,α_C,α_T).
Symmetric Dirichlet background parameters	A symmetric Dirichlet distribution is estimated.

Table 4.1: Features used in the **metamatti** classifier.

4.3 Results & Discussion

The main results in this chapter are threefold: the comparisons of the developed method with previous methods (Section 4.3.1), an independent validation of the performance with two large homeodomain datasets (Section 4.3.2), and a brief explanation of the publicly available implementation of the classification method (Section). Additionally, I also discuss the reasoning behind choosing an appropriate motif cluster count (Section 4.3.2.2), and compare the classifier to the more naive option of simply scoring motifs with average motifs derived from clustered, aligned motifs (Section 4.3.3).

4.3.1 Performance comparison with previous methods

Classification performance of **metamatti** was compared to two methods with a related goal: MotifPrototyper (Xing and Karp, 2004) which classifies motifs into four TRANSFAC superfamilies (zinc coordinated, helix-turn-helix, β-

scaffold,basic), and SMLR which classifies motifs into six major classes of TF domains (Cys_2His_2 and Cys4 zinc fingers, homeodomains, forkhead domains, basic helix-loop-helices and basic zipper domains) (Narlikar and Hartemink, 2006).

4.3.1.1 MotifPrototyper

Classification accuracy comparison shows that **metamatti** outperforms MotifPrototyper (Xing and Karp, 2004) (Figure 4.1) across all four TF domain superfamilies. The margin between the two methods is especially clear when one compares **metamatti** with the 'full' dataset classification made by Xing and Karp (2004), which contains all members of the four superfamilies in the TRANSFAC class, as opposed to the reduced 'major class' set which contains all motifs with at least 10 examples in the dataset. The **metamatti** classification was made with the full dataset.

There are several possible reasons for the substantial difference in performance. Firstly, the MotifPrototyper classification is made simply with a maximum a posteriori scheme: each TRANSFAC superclass corresponds to a MotifPrototyper model, and motifs are assigned to the superclass which has the highest maximal posterior probability to be generated by the corresponding MotifPrototyper. **metamatti** instead uses the metamotif densities as a features in a more sophisticated, discriminative random forest based classifier, which assigns the class labels to a motif. Secondly, the metamotif inference algorithm I developed is not constrained to a fixed motif family column count, unlike the algorithm utilised in MotifPrototyper which estimates model parameters from aligned motifs. The method by which motifs are aligned and trimmed to equal length is not specified by Xing and Karp (2004). Thirdly, training several metamotifs per motif family, **metamatti** also accounts for the fact that not all columns in motif families can be accurately expressed as a single column wise probability distribution. Instead, recurring patterns in a motif set can be generated by multiple potentially shorter familial metamotif components in my model. Furthermore, the metamotif estimation algorithm treats some motif columns as noise with a column background model, improving the capacity to find recurring patterns from sequence motif sets and reducing over-fitting of familial models due to reporting

weak or nonexistent recurring trends.

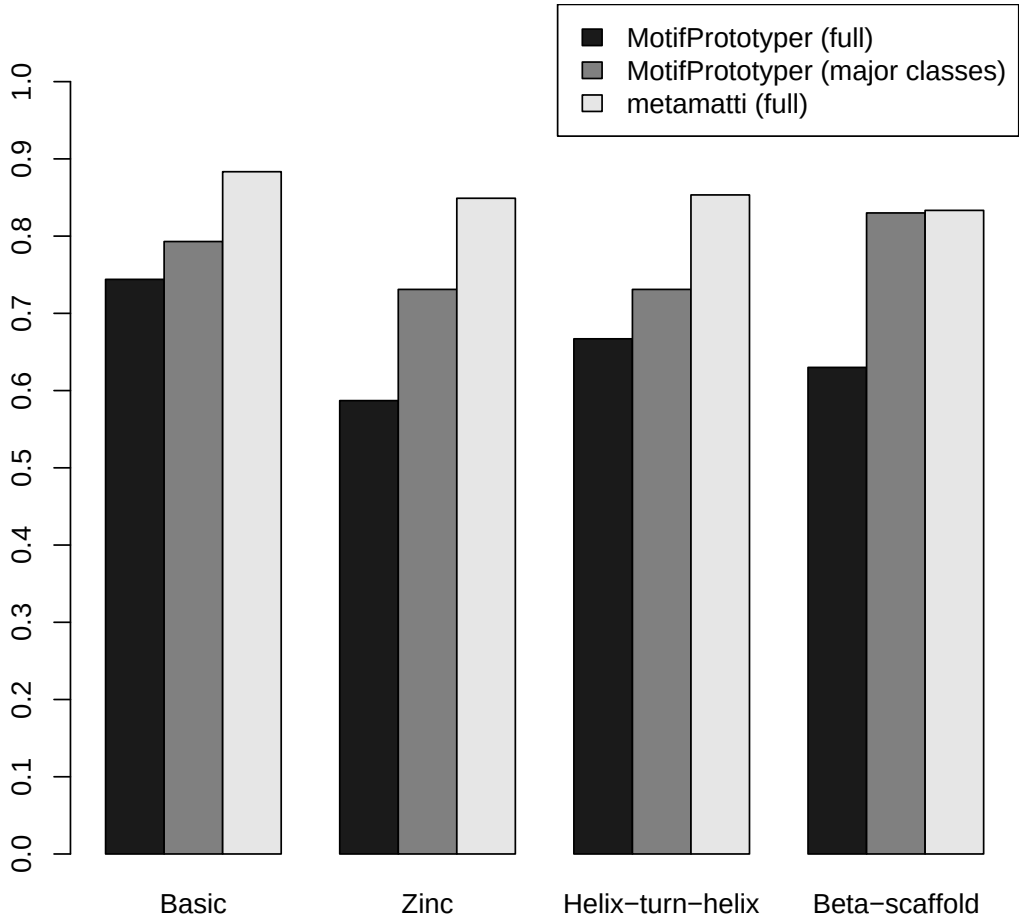

Figure 4.1: Accuracy comparison between TF domain superfamily level classification with **metamatti** and MotifPrototyper (10-fold crossvalidation). The 'major classes' refers to MotifPrototyper's reported performance for all motif families which include at least ten motif instances (Xing and Karp, 2004) in the TRANSFAC database (Matys et al., 2006) from the four superfamilies basic, zinc, helix-turn-helix and β-scaffold. 'Full' refers to a classification of all motifs in the four superfamilies, instead of just the major classes.

4.3.1.2 Sparse Multinomial Logistic Regression

To compare **metamatti** with SMLR (Narlikar and Hartemink, 2006), I conducted the TRANSFAC class level classification with the same subset of TRANSFAC 12.2 PWMs that were classified with SMLR. The overall classification accuracy comparison shows that **metamatti** has a marginally improved performance at 89.5% classification accuracy over the 87% reported for SMLR. The class-by-class accuracy figures (Figure 4.2) and the confusion matrix of the 6-way TRANSFAC motif family classifier (Table 4.3) however make it evident firstly that the ability of sequence motif properties to distinguish motifs by binding domain varies considerably depending on the domain both for **metamatti** and SMLR, and secondly that the higher classification accuracy comes at the cost of a 14% drop in the classification accuracy of the bHLH family (89% accuracy with SMLR, 75% with **metamatti**). The partially palindromic E-box motif CAGGTG appears to be the most common type misclassified in the erroneous bHLH motif cases. Inspection of family assignments of motifs in the TRANSFAC database shows that closely similar motifs with the CAGGTG consensus have been annotated with all of bHLH and C_2H_2 zinc finger families, highlighting a general limitation of a sequence PWM feature based motif family classification methods. Overall, the variability in accuracy across classes is not surprising: Luscombe and Thornton (2002) already describe sequence-specific DNA binding motifs into 'highly specific' (e.g. TATA binding protein and the basic zipper domain) and 'multi-specific' (e.g. homeodomain, C_2H_2 and Cys_4 type zinc finger domains), i.e. that different domains show different degree of constraint in the binding profiles seen in nature, which can make some domains harder to classify even with sophisticated methods. Random forest classification in fact outputs a classification probability for each of the potential classes. I in fact use this property of random forest classification in Chapter 5 (Section 5.3.6.5) to choose a confidence level for classification decisions, instead of reporting a class for all input motifs regardless of the uncertainty.

Motif family prediction methods ultimately rely on the structural mode of interaction by a protein DNA binding domain being reflected as a DNA sequence specificity pattern, and that pattern being distinct to each motif family as a

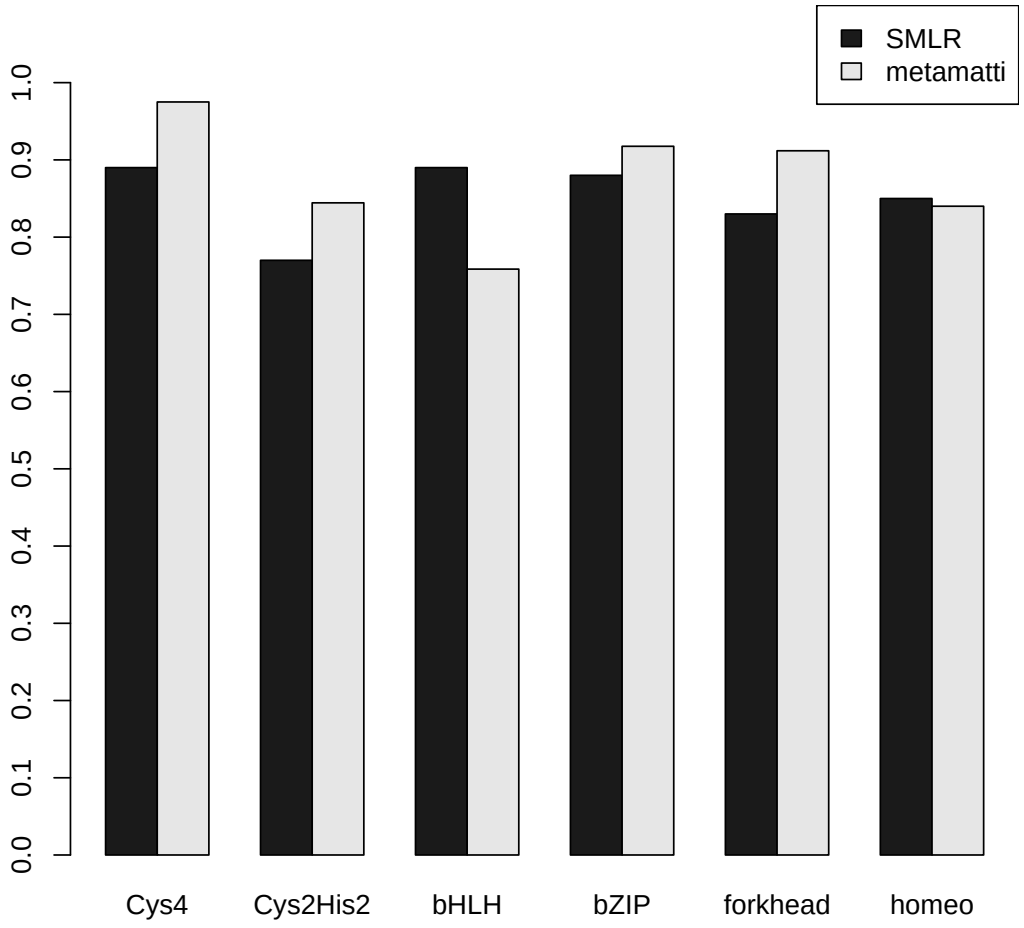

Figure 4.2: Accuracy comparison between the TF domain family classification with metamatti, and SMLR (k-fold cross-validation).

	Cys4	C2H2	bHLH	bZIP	Forkhead	Homeodomain	Class error
Cys4	39	0	0	0	0	1	**2.5%**
C2H2	0	38	3	0	1	3	**15.6%**
bHLH	0	2	22	5	0	0	**24.0%**
bZIP	0	3	0	78	0	4	**8.0%**
Forkhead	0	0	0	0	31	2	**9.0%**
Homeodomain	2	1	1	3	0	37	**16.0%**
Totals	**41**	**43**	**26**	**86**	**32**	**47**	

Figure 4.3: Confusion matrix of the 6-way TRANSFAC motif classification with the **metamatti** classifier. Columns correspond to the real class, and rows to the predicted class.

result of co-evolution of the two protein and its binding sites. As the above example of CANNTG sites shows, this is not always the case in nature: certain bHLH and Snail-like C_2H_2 like factors for example are thought to bind with closely similar specificities to compete for the same binding site positions (Nieto, 2002). The familial tendencies observed for DNA binding sites of transcription factors are thought to be due to both biophysical constraints on the possible DNA binding site patterns of a certain binding domain and evolutionary forces that maintain the familial DNA specificities distinct. Such forces range from functional redundancy of paralogous factors with overlapping binding sites (Kafri et al., 2005) to antagonistic regulation by opposing activators and repressors (Tanaka et al., 1993). To give an example of the inherent differences between TF domains, the C_2H_2 domain noted above has been found to be extremely plastic and a number of individual zinc fingers have even combined to very long (18bp) binding site patterns in a highly modular fashion (Dreier et al., 2001, 2000). In contrast, the bHLH domain has been observed to be much more strongly constrained in its DNA binding tendencies in a thorough mutagenesis study of the DNA contacting residues of the Max transcription factor (Maerkl and Quake, 2009). Further work is clearly needed to cover the full spectrum of binding site patterns explored by

sequence specific DNA binding domains, which also highlights the need for models such as the metamotif that describe recurring patterns in sequence motifs.

4.3.2 Performance measurement of two large homeodomain datasets

The previous motif classification work, which I compare my method with, has relied on cross-validation based estimation of classification accuracy from a single public database (Narlikar and Hartemink, 2006; Sandelin and Wasserman, 2004; Xing and Karp, 2004). Recent advances in protein-DNA interaction assaying have however resulted to the availability of several new experimental regulatory motif data sets that are not deposited in TRANSFAC. I wanted to assess the performance of **metamatti** with two homeodomain motif sets recovered from different species and via different experimental methods. The evaluation also allowed me to compare classification error rates achieved in independent datasets to the error rate predicted by **metamatti** classification for the homeodomain motif family. I applied **metamatti** to the *Mus musculus* PWMs constructed from the Berger et al. (2008) protein binding microarray motif data and reported the relative frequency at which the motifs were classified by **metamatti** with the homeodomain label (out of the six possible classes). Similarly, I classified motifs from the Noyes et al. (2008a) *Drosophila melanogaster* bacterial one-hybrid motif datasets.

The classification accuracy rates for both homeodomain motif sets were shown to be high, and in good agreement with the out-of-bag accuracy estimate of 91.3% reported by the **metamatti** random forest classifier during classifier training: 92.1% and 91.7% of the homeodomain motifs in the Berger et al. (2008) set of 84 motifs, and the Noyes et al. (2008a) set of 177 motifs, were correctly classified, respectively. I studied the misclassified examples from the *Drosophila melanogaster* homeodomain datasets in more detail to see where the misclassified motifs lie in the homeodomain specificity group clustering presented in Noyes et al. (2008a). Interestingly, the misclassifications were shown to be atypical homeodomains which do not contain the canonical TAATTA core and fall amongst the smaller specificity groups. The misclassified motifs included three

TGIF-Exd-like motifs (Vis, Hth, Exd), two Iroquois-like (Ara, Mirr), one Six-like (Optix) and an outlier from the specificity group clustering (Figure 4.4A). A similar trend of non-canonical homeodomains being primarily amongst the misclassified was also noted for the *Mus musculus* homeodomain motifs (4.4B). This is most likely explained by atypical homeodomain motifs not being well covered well by the TRANSFAC 12.2 training set; No closely matching homeodomain motifs were observed in TRANSFAC 12.2 to many of the misclassified motifs.

Figure 4.4: Misclassified homeodomain motifs in the A) Noyes et al. (2008a) and the B) Berger et al. (2008) datasets.

4.3.2.1 Classifying homeodomain motifs by their specificity group

I also wanted to test if a **metamatti**-like classifier could be trained to detect more detailed differences between motif groups than motif family or superfamily, a question which the previous methods have not addressed. I therefore labelled the *Drosophila melanogaster* homeodomain motifs with the homeodomain specificity

groups suggested by Noyes et al. (2008a) and estimated a single metamotif with `nmmetainfer` from each of the specificity groups. A single metamotif was used because of the small total number of motifs in the training data. I then trained a **metamatti** classifier with these metamotifs similarly as described above in Section 4.2. A remarkably high accuracy of 84% (confusion matrix shown in Table 4.5), when all Noyes et al. (2008a) homeodomain motifs with 3 or more examples per specificity group were included in the classification (9-way classification). The applicability of supervised machine learning strategies that aim to learn motif type labels more precise than the DNA binding domain family are however currently limited by the amount of available training data. For instance, the 84 motifs in the Noyes et al. (2008a) dataset contain examples of 11 specificity groups which are very biased to the two largest groups (Antennapedia and Engrailed, with 25 and 15 examples, respectively), with several specificity groups containing as few as two to four examples (Ladybird, Iroquis, NK-1, NK-2, TGIF-Exd, Bcd). This makes classifier error estimation imprecise especially for the weakly represented classes and results in the major classes, which have as much as eightfold as many examples present in the training dataset, to have considerable weight in predictions over the smaller classes (such as to maximise overall classification accuracy). Methods like **metamatti** can however become increasingly relevant once more high-throughput TF DNA specificity data becomes available.

	AbdB	Antp	Bar	Bed	Engrailed	Iroquis	NK-1	NK-2	TGIF-Exd	Class
AbdB	5	0	0	0	0	0	0	0	0	**0.00**
Antp	0	15	0	0	2	0	0	0	0	**0.12**
Bar	0	0	5	0	1	0	0	0	0	**0.17**
Bcd	0	0	0	4	0	0	0	0	0	**0.00**
Engrailed	0	1	1	0	23	0	0	0	0	**0.08**
Iroquis	0	0	0	0	0	3	0	0	0	**0.00**
NK-1	0	0	0	0	3	0	2	0	0	**0.60**
NK-2	0	0	0	1	0	0	0	2	0	**0.33**
TGIF-Exd	0	0	0	0	0	0	0	1	3	**0.25**
Totals	**5**	**16**	**6**	**5**	**29**	**3**	**2**	**3**	**3**	

Figure 4.5: Confusion matrix of the homeodomain specificity group classifier. Columns represent the real class, and rows represent the predicted class.

4.3.2.2 Clustering of motifs prior to metamotif training

Clustering of the motifs, and training metamotifs from motif clusters, was motivated by the requirement to choose a value for the metamotif count parameter of the metamotif inference algorithm, and to limit the metamotif search space. Inspection of clusters at cutoff 6.0 showed no clusters with more than three strongly distinct recurring patterns. Although for many motif clusters there were clearly less than three distinct recurring metamotif patterns present at the clustering cutoff of 6.0, the metamotif inference algorithm was found to treat these cases by either inferring closely similar duplicate metamotifs (such as metamotifs 1 and 2 in Figure 2.10A) or short metamotifs with mean nucleotide weights with low information content, or occasionally splitting the metamotif segments in several independent parts. This suggested that together with a sparse machine learning strategy such as a random forests, it would be advantageous to choose a high metamotif count that would describe the input motif set in as much detail as possible, with the price of some potentially redundant features in the feature set (densities for duplicate or low information metamotifs). I validated this assumption by retraining the classifier with two metamotifs per cluster (a total of 130 metamotifs). The classifier trained with two metamotifs per family resulted in a mild decrease in the classification accuracy (88.4%, as opposed to 89.5% with three metamotifs per cluster), suggesting that the additional metamotifs were indeed informative.

4.3.3 Comparing a metamotif density based classification to a Cartesian distance based classifier

I assessed the importance of the metamotif density score in the **metamatti** classifier by comparing it to a more naive classifier where we replace the metamotif average and maximum scores with average and maximum SSD distances computed between the training set motifs and 'average motifs' of each of the motif families. The average motifs used in the more naive classifier were the mean PWMs of the metamotifs trained with `nmmetainfer`. They were used for classification by scoring the training set motifs with an SSD distance metric with each of the metamotifs. We found that the classifier accuracy achieved with the

SSD metric was lower to the metamotif density based classifier by 1.4% (accuracy of 88.1%), suggesting that both the metamotif mean and the column wise precision values which contribute to the metamotif density scores are partially responsible for **metamatti**'s high performance. Furthermore, I tested training a classifier with cluster average motifs instead of the metamotif segments, resulting in an accuracy figure of 86.5%, suggesting that not only is the metamotif density a suitable score, but that the motif segments identified by the metamotif inference algorithm provide a classifier that generalises better than simply using average motifs inferred by clustering and collapsing clustered motifs to an average representation.

4.3.4 Making metamatti available

Once I had shown the favourable performance of **metamatti** with respect to previous related methods, it became important to make the classification method readily available. Much like with the familial PWM prior work described in the previous chapter, I wanted to make it usable for both experienced and inexperienced users, with as low a barrier to installing and using it as possible. The following sections describe two ways in which **metamatti** can be taken advantage of.

4.3.4.1 The metamatti R package

The metamotif based classifier was initially developed as a series of R and ruby scripts. Distributing the tool as an R package was therefore a natural choice. The R package can be used to predict using classifiers either packaged in the software (included as R datasets loadable with the `data()` function), or ones trained with the package based on training data. The classifier training procedure also optionally plots a precision-recall curve and a variable importance graph, similar to those shown in Chapter 5. Furthermore, the JASPAR based classifier noted in this example is introduced and applied in Chapter 5 (Section 5.3.6.5).

The package source code, installation instructions and documentation is available at `http://www.github.com/mz2/metamatti`. A brief usage example is provided below.

```
#Load the library
library(metamatti)

# Get a list of available metamatti classifiers
# alternatively way to accomplish this is:
# try(data(package="metamatti"))'
# Due to the licensing terms of the TRANSFAC database,
# the TRANSFAC based classifiers are not made publicly available.
# Additional classifiers can however be trained
# as shown below.
getAvailableMetamattiClassifiers()
#"transfac-class-6-way", "transfac-superclass-4-way", "jaspar-5-way"

# Extract features from your motifs of interest
features <-
  extractMetamattiFeatures("your-motifs.xms","jaspar-5-way")

# trainMetamattiForest(features,classifierName) can be used to
# train a new random forest classifier. Classifier training will
# also output a precision-recall graph
# (in this case jaspar-5-way-prec-recall.pdf,
# and a graph of variable importances
# ("jaspar-5-way-importances.pdf")
# in the working directory.
forest <- trainMetamattiForest(features,"jaspar-5-way")

# Alternatively, you can retrieve a jaspar-5-way classifier which is
# packaged alongside metamatti.
# Because the training sets are exposed as standard R datasets,
# you can also accomplish this with data("jaspar-5-way")'
forest <- getMetamattiForest("jaspar-5-way")

# Predict the class for the motifs
# Note that this is in fact a function from the randomForest package
# (the package is loaded upon loading the metamatti' library)
preds <- predict(features,forest)
```

4.3.5 The metamatti web server

In addition to the **metamatti** R package, I also created a simple web server application for motif family prediction. This was done most importantly because the outside dependencies required for installing the R package can act as a barrier of entry for inexperienced users, and because a web based application makes it possible to expose the TRANSFAC family classification to outside users (re-distributing the training data needed for it in the R package is impossible due to the licensing terms). The **metamatti** server can be used with a web browser (Figure 4.6) with a rather Spartan form based user interface. It also responds to a JavaScript Object Notation (JSON[1]) based response format to web service API calls. Documentation for using the web service API is included alongside the freely available (LGPL licensed) source code of the project at

[1] http://www.json.org

http://www.www.github.com/mz2/metamatti. It was implemented using Ruby on Rails (http://www.rubyonrails.org).

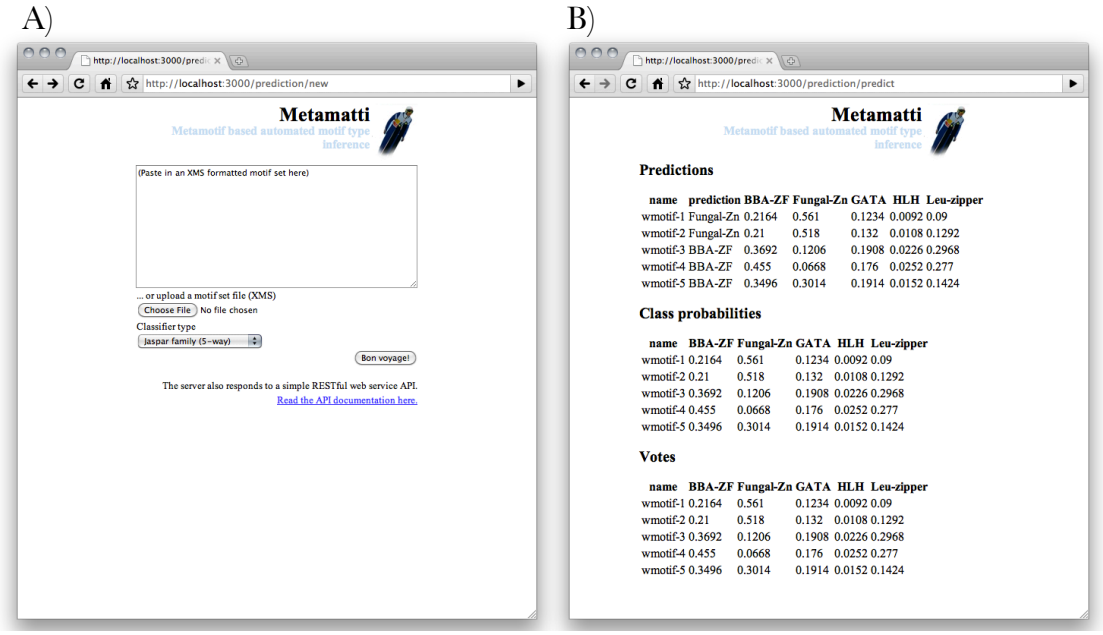

Figure 4.6: The **metamatti** motif classification web server. A) A screenshot of the prediction submission form. A motif set is entered either by pasting it to the form, uploaded as a file, or sent in a web service API call. B) A screenshot of the prediction report view. The tabular reports can be copied and pasted from (for instance to MS Excel), and they are also made available in a machine readable tabular (tab separated value) format through the web service API.