OPEN ACCESS Freely available online

PLoS one

A Bayesian Search for Transcriptional Motifs

Andrew K. Miller[1]*, Cristin G. Print[2], Poul M. F. Nielsen[1,3], Edmund J. Crampin[1,3]*

1 Auckland Bioengineering Institute, The University of Auckland, Auckland, New Zealand, **2** Department of Molecular Medicine and NZ Bioinformatics Institute, The University of Auckland, Auckland, New Zealand, **3** Department of Engineering Science, Faculty of Engineering, The University of Auckland, Auckland, New Zealand

Abstract

Identifying transcription factor (TF) binding sites (TFBSs) is an important step towards understanding transcriptional regulation. A common approach is to use gaplessly aligned, experimentally supported TFBSs for a particular TF, and algorithmically search for more occurrences of the same TFBSs. The largest publicly available databases of TF binding specificities contain models which are represented as position weight matrices (PWM). There are other methods using more sophisticated representations, but these have more limited databases, or aren't publicly available. Therefore, this paper focuses on methods that search using one PWM per TF. An algorithm, MATCHTM, for identifying TFBSs corresponding to a particular PWM is available, but is not based on a rigorous statistical model of TF binding, making it difficult to interpret or adjust the parameters and output of the algorithm. Furthermore, there is no public description of the algorithm sufficient to exactly reproduce it. Another algorithm, MAST, computes a p-value for the presence of a TFBS using true probabilities of finding each base at each offset from that position. We developed a statistical model, BaSeTraM, for the binding of TFs to TFBSs, taking into account random variation in the base present at each position within a TFBS. Treating the counts in the matrices and the sequences of sites as random variables, we combine this TFBS composition model with a background model to obtain a Bayesian classifier. We implemented our classifier in a package (SBaSeTraM). We tested SBaSeTraM against a MATCHTM implementation by searching all probes used in an experimental *Saccharomyces cerevisiae* TF binding dataset, and comparing our predictions to the data. We found no statistically significant differences in sensitivity between the algorithms (at fixed selectivity), indicating that SBaSeTraM's performance is at least comparable to the leading currently available algorithm. Our software is freely available at: http://wiki.github.com/A1kmm/sbasetram/building-the-tools.

Citation: Miller AK, Print CG, Nielsen PMF, Crampin EJ (2010) A Bayesian Search for Transcriptional Motifs. PLoS ONE 5(11): e13897. doi:10.1371/journal.pone.0013897

Editor: Diego Di Bernardo, Fondazione Telethon, Italy

Received May 31, 2010; **Accepted** October 20, 2010; **Published** November 18, 2010

Funding: This work was supported by the New Zealand Tertiary Education Commission [Top Achiever's Doctoral Scholarship] to AKM (http://www.tec.govt.nz); the New Zealand Health Research Council International Investment Opportunities Fund (http://www.hrc.govt.nz/) to CGP and EJC; the Breast Cancer Research Trust (http://www.breastcancercure.org.nz/) to CGP and EJC; and the New Zealand Foundation for Research, Science and Technology to CGP and EJC (http://www.frst.govt.nz/). This publication is based on work (by EJC) that was supported in part by award No. KUK-C1-013-04, made by King Abdullah University of Science and Technology (http://www.kaust.edu.sa/). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: ak.miller@auckland.ac.nz (AKM); e.crampin@auckland.ac.nz (EJC)

Introduction

Identifying which transcription factors bind to which promoters is an important step towards understanding the transcriptional regulatory code. This identification process can be divided into two parts: determining the binding specificity of specific transcription factors, and then identifying TFBSs in a sequence using the binding specificity information.

There have been a number of papers proposing methods for one or both parts of the problem. Methods for finding transcription factors (as motifs which are statistically over-represented in sequences) can be broadly classified as those based on phylogenetic footprinting, and those which are not. These methods have been widely compared [1,2] and reviewed [3]. The software implementations associated with many of these methods often also include software to use the motifs to identify TFBSs. For example, the popular motif finding software MEME [4] is packaged with the MAST software [5].

The link between determining binding specificity and finding sites where the transcription factor is likely to bind is the way in which binding specificity is represented. At present, the largest databases which are generally available, such as TRANSFAC [6] and JASPAR [7], represent binding specificity using an ungapped position weight matrix (PWM) representation. Each entry in an ungapped PWM, $\mathbf{F}$, is a weight for finding a particular base at a particular position from the start of the motif. There are several types of weights possible, but in this paper, we consider weights given as a raw count. At each aligned position i in the binding footprint, a frequency is recorded for each base b to give the matrix entry $\mathbf{F}_{ib}$. Let m denote the total number of aligned sequences. Not all TFBS sequences are aligned to both ends, and so for each i, $\sum_j F_{ij} \leq m$. Note that in algorithms such as MEME, the algorithm alternates between finding an alignment, and determining the PWM, until the algorithm meets a termination condition and the final PWM is produced.

There are more sophisticated representations for transcription factor binding specificity, such as the Hidden Markov Model (HMM) approach used by MAPPER [8]. However, TRANSFAC and JASPAR collectively include a reasonably large number of matrices, and these are available to the public (albeit under commercial terms in the case of TRANSFAC). Other databases are either smaller in size, or as in the case of MAPPER, binding models are not available to the public (Voichita D. Marinescu, personal communication). For this reason, the focus of this paper is

on methodology which uses only ungapped PWM information to search for transcription factor binding sites.

Representing transcription factor binding specificities in this form means that no data is stored on the interaction of binding specificity between different base positions in the binding sites. This is a reasonable approximation, as molecular binding models describing the interactions between transcription factors and DNA have shown that binding energies are approximately additive between bases [9] (in other words, interaction of binding specificity is negligible).

Existing PWM based search methodologies, such as MATCHTM, have not been justified based on a formal statistical model. MATCHTM instead computes scores using the formula [10]

$$\sum_{i=1}^{n}\left(\sum_{b\in\{A,C,G,T\}} F_{ib}\ln 4F_{ib}\right)F_{iB_i} \tag{1}$$

where n is the length of the matrix, B_i is the base at position i in the sequence, and F_{ib} is the frequency of base b at position i.

Methods

Overall approach

Let **A** be a sequence of bases of length at least n (where each base can be A, G, C, or T). We aim to make a decision about whether there is an exactly aligned TFBS starting at the beginning of **A**.

We define a TFBS as a locus that is under evolutionary pressure so the sequence is one that a particular transcription factor will bind to. The sequence is used as evidence supporting the hypothesis that there is a TFBS at a particular locus. For example, the presence of a sequence exactly identical to the consensus sequence for the transcription factor is strong evidence for a TFBS. A sequence which is more distantly similar to the consensus sequence is weaker evidence for there being a TFBS. This is because there are an increasing number of possible sequences as the deviation from the consensus sequence increases, and so the null hypothesis that similarity to the consensus sequence arose by chance (as opposed to natural selection) becomes more credible.

Under this definition, a transcription factor either binds to a TFBS, or it does not; there is no attempt to model the degree of affinity, only to determine if there is evidence for an underlying process. Note that evolutionary pressure may select for a moderate TF-TFBS affinity, but against a stronger affinity. In this case, evidence for the TFBS is reduced, but may still be enough to detect the site.

We use two models of putative TFBS sequences. The foreground model describes the distribution of sequences under the alternative hypothesis that there is a TFBS at the site. The background model describes the distribution of sequences under the null hypothesis that there is no TFBS at the site.

Foreground model

Our foreground model is best introduced in terms of a matrix of hidden parameters p_{ij} which represent the probability that a true TFBS will contain base j at position i. This parameter should not be confused with $\frac{F_{ij}}{\sum_k F_{ik}}$, which is merely an estimator of p_{ij}. The true p_{ij} is unknown. For this reason, we build a statistical model of F_{ij}, so we can express the joint distribution of F_{ij} and the TFBS sequence, under the alternative hypothesis. We refer to the alternative hypothesis that this model applies as H_1.

Our foreground model requires that each base in a TFBS is independently selected in accordance with the hidden parameters. In practice, there are two ways in which new TFBSs are likely to arise. They may arise from convergent evolution, in which case the TFBS sequence is independent of all other TFBSs. Alternatively, an existing TFBS could be copied in a duplication event, creating a paralogous TFBS which is not independent of the original. Over time, however, mutations to less strongly conserved bases in the two TFBSs will reduce this dependence. For this reason, the independence assumption is reasonable except for very recently duplicated TFBSs.

If B_i is the base at position i into a TFBS, the probability of the sequence **B** is

$$P(\mathbf{B}|\mathbf{p}) = \Pi_i p_{iB_i} \tag{2}$$

We assume, under this same model, that $\mathbf{F}_i$ is a random variable produced by aligning n_i independent sequence samples (where $n_i = \sum_j F_{ij}$), and therefore that

$$F_{ij} \sim \text{Binomial}(n_i, p_{ij}) \tag{3}$$

Hence,

$$P(F_{ij} = f_{ij}|p_{ij}) = \binom{n_i}{f_{ij}} p_{ij}^{f_{ij}}(1-p_{ij})^{n_i - f_{ij}} \tag{4}$$

where f_{ij} is a non-negative integer representing a frequency.

Now,

$$\begin{aligned} &P(B_i = b \cap F_{ib} = f_{ib}|p_{ib}) \\ &\qquad = \binom{n_i}{f_{ib}} p_{ib}^{f_{ib}}(1-p_{ib})^{n_i - f_{ib}} p_{ib} \end{aligned} \tag{5}$$

$$\begin{aligned} P(B_i = b|F_{ib} = f_{ib}) &= \frac{P(B_i = b \cap F_{ib} = f_{ib})}{P(F_{ib} = f_{ib})} \\ &= \frac{\int_0^1 \binom{n_i}{f_{ib}} p_{ib}^{f_{ib}}(1-p_{ib})^{n_i - f_{ib}} p_{ib} dp_{ib}}{\int_0^1 \binom{n_i}{f_{ib}} p_{ib}^{f_{ib}}(1-p_{ib})^{n_i - f_{ib}} dp_{ib}} \\ &= \frac{\beta(f_{ib}+2, n_i - f_{ib}+1)}{\beta(f_{ib}+1, n_i - f_{ib}+1)} \end{aligned} \tag{6}$$

where $\beta(x,y)$ is the Euler Beta function [11].

Note that we assume that $P(p_{ib}) = 1$ (*i.e.* without any samples from f_{ib}, we know nothing about p_{ib}). This is the same as the $\text{Beta}(1,1)$ distribution, from the conjugate prior family to the Binomial distribution.

This gives us the ability to compute the probability of a given sequence under the alternative hypothesis:

$$P(\mathbf{B}|\mathbf{F}) = \Pi_{i=0}^{l} \frac{\beta(f_{iB_i}+2, n_i - f_{iB_i}+1)}{\beta(f_{iB_i}+1, n_i - f_{iB_i}+1)} \tag{7}$$

Background model

We used a simple first-order Markov chain model, with one parameter for each base, q_b, describing the probability that the base b occurs at a particular point in the sequence. In addition, we introduce one parameter, $t_{b_1 b_2}$ for each pair of bases (b_1,b_2), describing the conditional probability of finding base b_2 at a point in the sequence, given that b_1 was present one base-pair earlier in the sequence. We refer to the null hypothesis that this model applies as H_0.

We will assume that the foreground and background model are complementary. This is an approximation, because sequences might have higher order interactions not explained by either the foreground or background models. Making a simplifying assumption here is unavoidable because of the high complexity of these higher order interactions. For example, polypeptide coding sequences are considered background, and the distribution of the sequence of bases is determined by the effect of the polypeptide sequence on evolutionary fitness; something which would require more knowledge about biological function than is available, and is too complex to include in the background model.

However, the model nevertheless provides a principled approach for correcting for the length of the sequence, and for differences in the frequency of bases or pairs of bases. Hence,

$$P(\mathbf{B}|H_0) = q_{B_1} \Pi_{i=2}^{n} t_{B_{i-1} B_i} \tag{8}$$

Recall that B_i is the ith nucleotide in the sequence being tested for a motif occurrence.

Combining the models

In order to combine the foreground and background models, we start with Bayes' theorem:

$$P(H_1|\mathbf{B}) = \frac{P(\mathbf{B}|H_1)P(H_1)}{P(\mathbf{B})} \tag{9}$$

We assume the foreground and background models are complementary, so

$$P(\mathbf{B} \cap H_1) + P(\mathbf{B} \cap H_0) = P(\mathbf{B} \cap (H_0 \cup H_1)) = P(\mathbf{B}) \tag{10}$$

$$P(\mathbf{B}) = P(\mathbf{B}|H_1)P(H_1) + P(\mathbf{B}|H_0)P(H_0) \tag{11}$$

Due to complementarity,

$$P(H_0) = 1 - P(H_1) \tag{12}$$

This leaves the prior probability $P(H_1)$ as the only remaining unknown. This should be an estimate of the rate of occurrence of the TFBS in the genome (or other set of sequences being searched). As this is not known, we make a plausible assumption about $P(H_1)$, and later discuss the sensitivity of the accuracy of the method to this parameter.

We note that this combination of foreground and background models is able to represent a number of features to the extent that the information is present in the raw counts matrix. For example, gaps in the sequence correspond to regions in which the foreground is indistinguishable from the background, in which the value of p_{ib} is identical to the probability of finding the base in the background. Similarly, palindromes can be represented merely by the incorporation of the palindromic pattern into **p**. For this reason, there is no need for any special steps to be taken to allow BaSeTraM to find gapped or palindromic TFBS.

Comparison with other work

Our model shares some similarities with the model used in a previous study [12]. However, we have taken a different approach at a number of points, as we discuss below. The most notable benefit of our approach compared to the Bayesian approach presented in the paper is that the approach of Lähdesmäki et al. requires a computationally expensive Markov Chain Monte Carlo (MCMC) procedure, while we can efficiently compute the posterior probability for a given motif being at a given position.

One major difference between the two approaches is that Lähdesmäki et al. aims to identify the posterior probability of alignments of one or more motifs in a given promoter region, while BaSeTraM computes the probability that a single motif is found at a given site, and uses this to annotate a sequence with probable sites. Another difference is that BaSeTraM does not take into account uncertainty in the background probabilities (and instead focuses entirely on the uncertainties in frequencies in the foreground model). This approximation can be justified by the large quantity of data available to build the background model (as opposed to the foreground models), and the correspondingly low estimator variance. Using this simpler background model allows BaSeTraM to efficiently use a context-dependent background model.

In addition, Lähdesmäki et al. used a different derivation, by representing all foreground model frequencies at each position using a four-way multinomial distribution across all bases. In this paper we instead use a binomial distribution, where one Bernoulli outcome is that a base at position i used to build the PWM row $\mathbf{F}_i$ matches the base B_i, and the other is that it does not. In other words, we build a model of the motif matrix specific to **B**, while Lähdesmäki et al. builds a general model. As discussed in the Implementation section, our formulation allows us to find a computationally efficient closed form solution (dependent on pre-computed values of the β function) for the posterior probability.

Implementation

We developed an implementation, SBaSeTraM, of the Bayesian search method, BaSeTraM, described above. We also implemented the method described in [10], and refer to the implementation as GMATIM. As the implementation of MATCHTM provided by the authors of that paper is closed source, GMATIM may differ from the BioBase MATCHTM implementation. For example, that paper stated that "the core of each matrix is defined as the first five most conserved consecutive positions of a matrix". However, we have been unable to determine how the level of conservation of each group of 5 consecutive positions is measured and compared. To resolve this issue, we implemented GMATIM to simply find the 5 most conserved positions, where conservation at position i is measured as $\max_b f_{ib}$.

In addition, we have created a wrapper, called WrapMAST, around the stand-alone MAST [5] binary, which we built from the MEME 4.4.0 source code (downloaded from http://meme.sdsc.edu/ on the 2nd of July, 2010). WrapMAST converts matrices from TRANSFAC into the form produced by MEME. This involves converting the matrix of frequencies to a matrix of log-odds **L**. We have used the same formula used in the MEME software (using a background proportion of 0.25 for each base):

$$L_{ij} = \log_2 (4\hat{p}_{ij} + 10^{-200}) \tag{13}$$

$$\hat{p}_{ij} = \frac{F_{ij}}{\sum_k F_{ik}} \tag{14}$$

The addition of 10^{-200} is used (as in MEME) in ensure that L_{ij} has a real value even when $\hat{p}_{ij} = 0$. For each PWM, WrapMAST invokes MAST in hit list mode to search all probes. It then parses the output from MAST and outputs them in the same format used by SBaSeTraM (but with the p-value from MAST used in place of the posterior probability from SBaSeTraM).

SBaSeTraM, GMATIM, and WrapMAST are written in Haskell, and we have aimed to make the source code of each program a succinct and readable description of the corresponding algorithm. SBaSeTraM, WrapMAST, and GMATIM provide a similar command line interface (and share common code), so as to simplify the design of analyses which compare the algorithms.

Due to the possibility of numerical underflow from very small probabilities, our SBaSeTraM and GMATIM implementations make use of log probabilities (base e).

It is necessary for SBaSeTraM to compute the posterior probability, $P(H_1|\mathbf{B})$, at every position in the sequence being searched, for every TFBS matrix (with the exception that there is no search for TFBS matrices of length l in a sequence of length n at starting positions $i > n-l$). For this reason, it is important that the posterior probability can be computed efficiently.

The β function has no closed form, and needs to be calculated numerically. To avoid expensive computations in our inner loop, for each matrix, we pre-compute $v_{ib} = \ln \frac{\beta(f_{ib}+2, n_i - f_{ib}+1)}{\beta(f_{ib}+1, n_i - f_{ib}+1)}$ for each i and b. We also pre-compute all partial sums of the series $0 + \sum_{i=1}^{N} \ln t_{B_{i-1}B_i}$, where N is the length of the sequence $\mathbf{B}$. Let s_i be the ith entry in the series, so,

$$s_1 = 0 \tag{15}$$

$$s_2 = \ln t_{B_1 B_2} \tag{16}$$

$$s_3 = \ln t_{B_1 B_2} + \ln t_{B_2 B_3} \tag{17}$$

and so on. This means that:

$$\ln P(B_i \cap ... \cap B_{i+l-1} | H_{0,B_i}) = \ln q_{B_i} + s_{i+l-1} - s_i \tag{18}$$

$$\ln P(B_i \cap ... \cap B_{i+l-1} | H_{1,B_i}) = \sum_{j=i}^{i+l-1} v_{j-i+1,B_j} \tag{19}$$

Note that equation 18 is a log-transformed equivalent of equation 8, and similarly, equation 19 is a log-transformed equivalent of equation 7.

$$\begin{aligned} &\ln P(H_{1,B_i} | B_i \cap ... \cap B_{i+l-1}) \\ &= \ln P(B_i \cap ... \cap B_{i+l-1} | H_{1,B_i}) + \ln P(H_1) - \\ &\quad \text{logsumexp}(\ln P(B_i \cap \ldots \cap B_{i+l-1} | H_{1,B_i}) + \ln P(H_1), \\ &\quad\quad \ln P(B_i \cap \ldots \cap B_{i+l-1} | H_{0,B_i}) + \ln P(H_0)), \end{aligned} \tag{20}$$

where $\text{logsumexp}(x,y)$ is a function which computes $\ln(e^x + e^y)$ while avoiding numerical underflow for large magnitude negative values of x and y, by computing $a + \ln(e^{x-a} + e^{y-a})$ for $a = \max(x,y)$.

We compute the vector q and the matrix t once, across all nucleotide sequences to be processed, by counting the number of occurrences of each base and sequence of the two bases, respectively, and dividing by the pooled total number of occurrences.

For each site, we compute the log-posterior probability and test it against a cut-off (as discussed below) to decide whether the TFBS occurs at that site. We search for sites, both on the sequences provided, and on the reverse complement of those sequences.

We retrieved the online supplement for [13] at http://fraenkel.mit.edu/Harbison/release_v24/. This data describes which of 6725 probes each of 182 different transcription factors bound to in a series of chromatin immunoprecipitation microarray (ChIP-chip) experiments. These probes were between 47 and 2764 base pairs long, with 95% between 92 and 1317 base pairs, 50% between 227 and 647 base pairs, and a median length of 359 base pairs. We also downloaded all TRANSFAC Saccharomyces Module matrices (TSM; [6]), as of 2009-06-16, from http://tsm.bioinf.med.uni-goettingen.de/.

Where a matrix used estimated rather than raw counts, as indicated by the occurrence of a decimal point in the 'frequency' matrix, that matrix was excluded (as we have assumed that raw counts will be used).

We filtered the set of probes, based on the experimental data, to only include those to which a transcription factor bound (for which we had a corresponding PWM). This left 1259 probes.

We then used each method to search the entire set of probes for TFBSs corresponding to each matrix, across all positions in the probe. Where the method detected the occurrence of a TFBS for a particular TF at any position in a probe, a positive result for that TF-probe combination was recorded. If no TFBSs were found at any position for a given TF a negative result was recorded. These results were then compared against the 'gold standard' experimental data. Only TFs which had corresponding matrices in TSM, and were also in the experimental results, were included.

We classified each included TF-probe pair into 4 categories:

- True Positive (TP) - positive prediction, and experimental determination of TF-probe interaction;
- False Positive (FP) - positive prediction, but no experimental determination of TF-probe interaction;
- True Negative (TN) - negative prediction, and no experimental determination of TF-probe interaction;
- False Negative (FN) - negative prediction, but experimental determination of TF-probe interaction.

In this paper we have used 0.001 as an approximation of the prior probability, because this value is credible as a frequency of occurrences. To determine the sensitivity of this parameter, we tested values that were one order of magnitude higher, and one and two orders of magnitude lower. The posterior probabilities obtained from doing this are increased or decreased, respectively, but once this is taken into account when selecting cut-offs, there is very little difference in the results within this range of prior probability parameters.

Results and Discussion

There were 38 different transcription factors in TRANSFAC Saccharomyces Module, of which 32 were made up of raw counts.

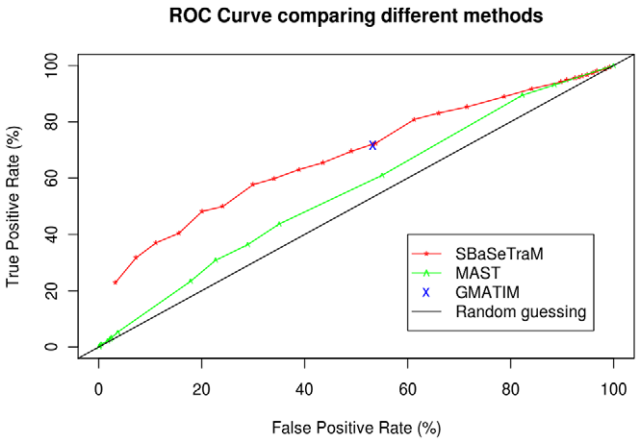

Figure 1. A receiver operating characteristics (ROC) curve comparing SBaSeTraM, GMATIM, and MAST. For SBaSeTraM, the posterior cut-off was varied to obtain a series of points. For MAST, the p-value cutoff was varied. For GMATIM, the parameters listed in the MATCHTM paper were used to generate the point on the curve.
doi:10.1371/journal.pone.0013897.g001

Of these, 16 were also found in the ChIP-chip dataset. These were tested against the 1259 different probes in the chromatin immunoprecipitation experiment. This gives 20144 different TF-probe pairs where we can classify whether the TF binds to the probe, and then check the classification. These results are shown in Figure 1.

We generated a ROC curve (Figure 1) for SBaSeTraM, by varying the posterior probability cut-off, and hence the trade-off between sensitivity and selectivity.

The point on the ROC curve generated using the parameters from [10] with GMATIM appears slightly below the ROC curve for BaSeTraM (GMATIM has 71.61% true positive rate for a 53.27% false positive rate). We found a posterior cutoff that generates a FPR close to this (with a posterior probability cut-off of 0.407, BaSeTraM achieved a 72.07% TPR at a FPR of 53.25%). At this point, we tested for a significant difference in the proportion of predictions which were correct; that is, $\frac{TP+TN}{TP+TN+FP+FN}$. We performed a comparison of these two binomial proportions, using the prop.test function in R [14], and obtained a one-sided p-value of 0.4603 (*i.e.* not significant to a 95% confidence level).

SBaSeTraM outperforms MAST when used through Wrap-MAST. It is worth noting that MAST is not typically used with TRANSFAC PWMs, and usually, multiple PWMs are used for each TF, and so the results cannot be used to make inferences about how well MAST performs together with MEME. The results do, however, illustrate the benefit of methods which take into account uncertainty in the foreground model.

We also carried out an analysis to see whether any particular TFs were making a large contribution to the overall prediction accuracy at this point. Figure 2 shows the differences between the two methods in the ROC space for each TF PWM. For each transcription factor, we have plotted an arrow from the point in the ROC space corresponding to the results for SBaSeTraM, to the point corresponding to the results from GMATIM. Some of the predictions are quite different; for example, for ADR1, SBaSeTraM found no occurrences, while GMATIM made numerous predictions, resulting in a true positive rate of 91.3% and a false positive rate of 96.0% (putting the accuracy for that particular TF below the line of no-discrimination). There was only one TF, GAL4, for which SBaSeTraM fell below the line of no-discrimination (which GMATIM predicted with a 17.4% true positive rate and a 0.8% false positive rate), and three TFs for which GMATIM fell below the line of no-discrimination (all of which were above or on the line of no-discrimination for SBaSeTraM). Unlike for SBaSeTraM, GMATIM predictions for HSF1, ROX1, and STE12 had true and false positive rates approaching 100%.

We also analysed the spread of true and false positive rates for each method. Figure 3 shows box-and-whisker plots for the true

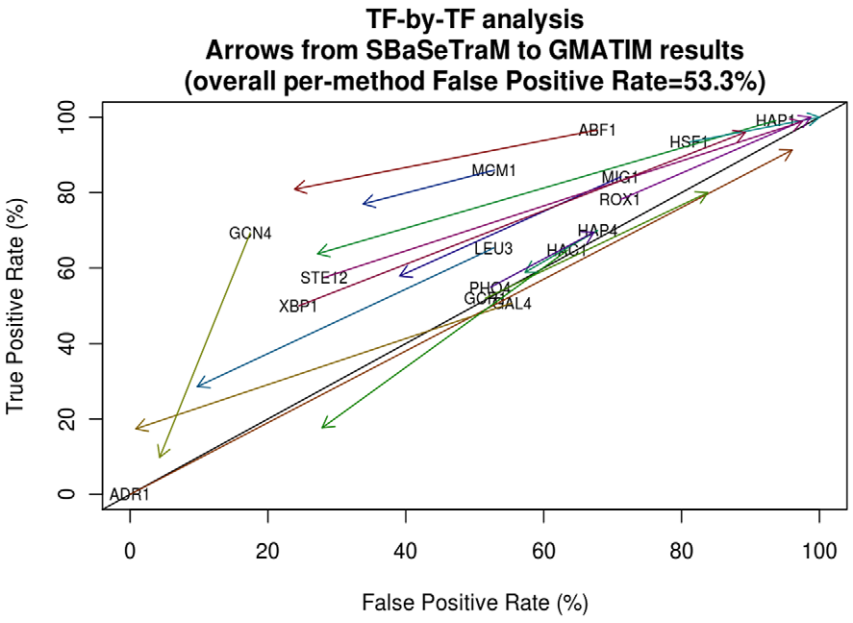

Figure 2. Comparing SBaSeTraM to GMATIM predictions for each transcription factor. The results are shown with the overall False Positive Rate for SBaSeTraM matched at that obtained from GMATIM with the parameters in the MATCHTM paper, namely 53.3%. Arrows run from the point obtained using SBaSeTraM to the point obtained using GMATIM.
doi:10.1371/journal.pone.0013897.g002

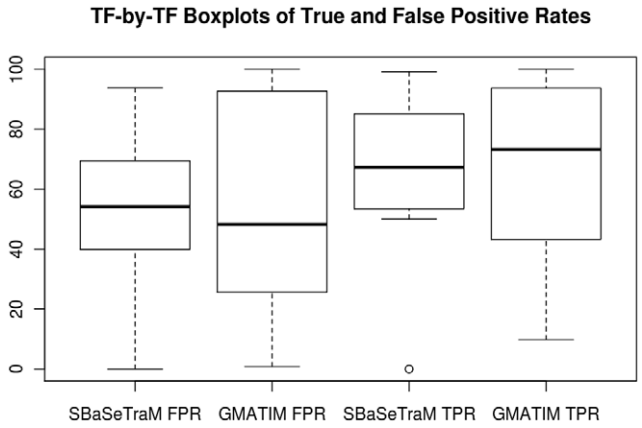

Figure 3. Box and whisker plot showing the spread of true and false positive rates for SBaSeTraM and GMATIM. The results are shown with the overall False Positive Rate for SBaSeTraM matched at that obtained from GMATIM with the parameters in the MATCHTM paper, namely 53.3%.
doi:10.1371/journal.pone.0013897.g003

and false positive rates for SBaSeTraM and GMATIM. Notably, there is a much greater distance between the upper and lower quartiles in both the true and false positive rates for GMATIM than there is for SBaSeTraM. This demonstrates that the BaSeTraM algorithm is more consistently controlling the trade-off between sensitivity and selectivity for each individual TF.

In addition, we used the bisection method to find a separate posterior probability cutoff for each of the 16 TFs that gave the SBaSeTraM method a FPR (for that TF) close to the FPR obtained with GMATIM. We allowed the method to terminate when a cutoff was found that brought the L_1 distance of the two FPRs within 0.001%, when an increase in cutoff resulted in an increased FPR (or a decrease in the cutoff resulted in a decrease in the FPR), or when no improvement in FPR was achieved after 4 iterations of the algorithm. The latter two conditions are necessary because there are a finite number of probes (1259), and there is no guarantee that there will be a cutoff which brings the SBaSeTraM FPR within 0.001% of the GMATIM FPR. In practice, for 8 of the 16 TFs, the difference between the final FPRs for the two methods was less than 0.001%, for 11 it was within 0.25%, and for 13 was within 0.5%. For HAC1, the final SBaSeTraM FPR was 0.966% higher than the GMATIM one, for XBP1 the GMATIM FPR was 1.158% higher, and for HAP1, the final SBaSeTraM FPR was 2.012% higher.

Using the same methodology used on the entire dataset (as discussed above), we tested for a statistically significant difference in proportion of predictions which were correct for each transcription factor, between GMATIM and SBaSeTraM (with the posterior probability cutoffs discussed in the previous paragraph). We obtained only one result where the p-value was less than 0.05, for GCN4 ($p = 0.00886$). For this TF, the FPR for both methods was 4.288%, the TPR for SBaSeTraM was 38.037%, while it was 9.816% for GMATIM. When we applied the Holm-Bonferroni procedure for multiple comparisons [15], none of the TF-by-TF results were significant to a 5% familywise error rate (FWER).

Conclusions

We have developed a Bayesian classifier for identifying TFBSs, which performs comparably to an existing algorithm, but which has a more principled statistical explanation, so that the trade-off between sensitivity and selectivity can be trivially adjusted, and the method can be altered to use different background models.

It is clear that the two methods are very similar in overall performance, and there is insufficient data in TSM to tell the two apart. The 95% confidence interval for the difference of the proportion correctly classified above runs from SBaSeTraM being 1.03% better, to GMATIM being 0.93% better. We therefore conclude that until there is more evidence that one method is better, from a performance standpoint, the two methods can be used interchangeably.

However, the fact that the statistical interpretation of BaSeTraM has been explained in rigorous terms, combined with the ease with which the posterior probability cut-off can be adjusted (as opposed to needing to adjust two separate parameters and re-run the analysis) makes the use of BaSeTraM preferable for many applications.

We note that despite the similarity in accuracy, the predictions made are not all the same; only 62.8% of all predictions of transcription factor binding made by SBaSeTraM with this posterior probability cut-off were also made by GMATIM.

The BaSeTraM statistical model includes a background model to be used. While a relatively uninformative background model is useful with the synthetic probes used in ChIP-chip analyses, using a different background model is likely to be important on genomic scale data, where there are localised variations in base frequencies.

When dealing with genomic scale data, it is also important that computation is reasonably efficient. It is also preferable that this computation can occur on modest hardware, so it is usable by groups without access to high-performance computing infrastructure.

In order to achieve these goals, we also developed a C++ implementation of BaSeTraM, called CBaSeTraM, which we optimised for the AMD64 architecture. We used Callgrind [16] to identify places where cache misses were occurring. We then used a customised allocator to ensure that all data which is needed in the inner loop (which is executed for each matrix for each alignment for each position) does not result in any cache misses, due to it being present in one cache page. As reading the level 1 and 2 caches are approximately 10 and 300 times faster than RAM, respectively, this leads to significant speed-ups. In this tool, we also implemented a sliding window determination of background model parameters q_b and $t_{b_1 b_2}$. Our implementation supports two distinct sliding windows; the intention is that one window is much larger than the other. The final estimate of each q_b and $t_{b_1 b_2}$ is the geometric mean of the two estimates. By default, the small window is 501 BP wide, and the large window is 2001 BP wide. Both windows are centred on the same base, which is used as the first position when testing for TFBSs. In addition, CBaSeTraM can use MPI [17] to search multiple sequences in parallel.

GMATIM, SBaSeTraM, and CBaSeTraM, as well as the programs used to test the methods, are Free/Open Source software. Instructions for building these programs are included as an online supplement.

Acknowledgments

The authors would like to thank the two anonymous reviewers for providing helpful feedback on this manuscript.

Author Contributions

Conceived and designed the experiments: AKM CP PMFN EJC. Performed the experiments: AKM. Analyzed the data: AKM. Contributed reagents/materials/analysis tools: AKM. Wrote the paper: AKM CP PMFN EJC.

References

1. Tompa M, Li N, Bailey T, Church G, De Moor B, et al. (2005) Assessing computational tools for the discovery of transcription factor binding sites. Nature Biotechnology 23: 137–144.
2. Mahony S, Auron P, Benos P (2007) DNA familial binding profiles made easy: comparison of various motif alignment and clustering strategies. PLoS Comput Biol 3: e61.
3. Das M, Dai H (2007) A survey of DNA motif finding algorithms. BMC Bioinformatics 8: S21.
4. Bailey T, Williams N, Misleh C, Li W (2006) MEME: discovering and analyzing DNA and protein sequence motifs. Nucleic Acids Research 34: W369.
5. Bailey T, Gribskov M (1998) Combining evidence using p-values: application to sequence homology searches. Bioinformatics 14: 48.
6. Matys V, Fricke E, Geffers R, et al. (2003) TRANSFAC: transcriptional regulation, from patterns to profiles. Nucleic Acids Research 31: 374.
7. Sandelin A, Alkema W, Engstrom P, Wasserman W, Lenhard B (2004) JASPAR: an open-access database for eukaryotic transcription factor binding profiles. Nucleic Acids Research 32: D91.
8. Marinescu V, Kohane I, Riva A (2005) MAPPER: a search engine for the computational identification of putative transcription factor binding sites in multiple genomes. BMC Bioinformatics 6: 79.
9. Liu L, Bader J (2007) Ab initio prediction of transcription factor binding sites. In: Pac. Symp. Biocomput. volume 12. pp 484–95.
10. Kel A, Gossling E, Reuter I, Cheremushkin E, Kel-Margoulis O, et al. (2003) MATCHTM: a tool for searching transcription factor binding sites in DNA sequences. Nucleic acids research 31: 3576.
11. Olver FWJ, Lozier DW, Boisvert RF, Clark CW (2010) NIST Handbook of Mathematical Functions - 5.12. Beta Function. Cambridge University Press. URL http://dlmf.nist.gov/5.12.
12. Lähdesmäki H, Rust A, Shmulevich I (2008) Probabilistic inference of transcription factor binding from multiple data sources. PLoS One 3.
13. Harbison C, Gordon D, Lee T, Rinaldi N, MacIsaac K, et al. (2004) Transcriptional regulatory code of a eukaryotic genome. Nature 431: 99–104.
14. R Development Core Team (2009) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org. ISBN 3-900051-07-0.
15. Holm S (1979) A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics 6: 65–70.
16. Weidendorfer J (2008) Sequential Performance analysis with Callgrind and KCachegrind. In: Tools for High Performance Computing: Proceedings of the 2nd International Workshop on Parallel Tools for High Performance Computing, July 2008, HLRS, Stuttgart. Springer. 93 p.
17. Gropp W, Thakur R, Lusk E (1999) Using MPI-2: Advanced features of the message passing interface. MIT Press Cambridge, MA, USA.