

Chapter 3. Modeling of transcription start sites

My initial objective here was to build a system which can predict transcription start sites in bulk genomic DNA sequences. For this problem, selectivity is vitally important: of all the possible positions in the human genome – around 3 billion in total – only a tiny proportion are expected to be actual transcription start sites, so even a low rate of overprediction, taken on a position-by-position basis, could still lead to a extremely large number of false positives across the genome as a whole, giving results which are of little value either to laboratory researchers wishing to perform directed promoter-mapping experiments, or as a starting point for performing other computational analyses.

There have been a number of attempts to develop computational methods of promoter prediction (see page 17). Older methods, predating the availability of large volumes of vertebrate genome sequence, tended to suffer from limited selectivity [Fickett and Hatzigeorgiou 1997, Scherf *et al.* 2000]. A more recent method, PromoterInspector [Scherf *et al.* 2000], aimed to provide a rather higher degree of selectivity, but it makes predictions for regions of the genome, giving an approximate area for the promoter but little specific information about the actual transcription start site. This program also has rather limited use for genome annotators, since it is a proprietary product. While a free web-based interface exist, users are only allowed to run a small number of analyses each month.

In order to develop a new prediction method, I wished to build a realistic model, preferably probabilistic in nature, of the sequence around transcription start sites. As well as providing a valuable predictive tool, using suitable machine learning methods to build such a model can also provide additional insights into the structures being investigated. Modeling approaches

are popular in sequence bioinformatics: in particular, Hidden Markov Models (HMMs) are widely used [Durbin *et al.* 1998], for example in the field of gene prediction where models such as Genscan represent the state of the art [Burge and Karlin, 1997]. In simple terms, HMMs can be viewed as two components. Firstly, the model architecture consists of a set of states and valid transitions between them. Secondly, a parameterization gives actual transition and emission probabilities. Given a specific architecture, the Baum-Welch optimization method offers a straightforward algorithm for finding a parameterization to optimally fit a given dataset, but learning architectures is much harder. HMMs such as the Genscan gene model have rather complex architectures, dedicated to a particular task and consciously designed and tested for this purpose by the method's developers. A less specialized approach is offered by profile HMMs, such as those used to build the Pfam protein family database [Bateman *et al.* 2002]. These have architectures built from simple repeating units. The Pfam models are built from multiple sequence alignments, and the “backbone” of states in the profile HMM represent the consensus of this alignment. This approach works very well for protein families, where all the members are evolutionarily related to one another, generally *via* point mutations and relatively small insertions and deletions, but profile models cannot be considered as a truly general approach to sequence analysis.

In this case, I decided that basic HMM methods were not an optimal approach for modeling promoters. Based on existing knowledge of transcription initiation (see chapter 1), promoters appear to be loosely-connected sets of motifs, rather than evolutionarily-related variants on a single theme, which suggests that Pfam-like profile models are not applicable. Other forms of HMM could potentially be designed which modeled motifs individually, and left some flexibility in their positions, and indeed steps have been taken in this direction by the Meta-MEME program [Grundy *et al.* 1997]. However, it is difficult, for example, to model the case of two motifs which must occur together, but in either order (and might perhaps even overlap). Moreover, it is likely that any non-trivial model architectures would have to be built either by hand or using heuristic methods, which potentially constrains the range of architectures which could viably

be explored.

I wished to search for interesting signals in as unconstrained a fashion as possible. I therefore set the following requirements for a new method, to be used for modeling transcription start sites (and hopefully extensible to other significant sites in biological sequence data):

- The model should be applicable to individual points (in their context) in a sequence, making accurate location of point features such as transcription start sites possible.
- The model should be modular, and built up from specific signals, rather than simply treating a whole sequence as a monolithic entity (as a profile HMM does).
- Any “architectural” aspects of the model must be learned automatically from supplied training data, preferably in the same process as the learning of individual signals, rather than a two-step process like Meta-MEME.

This set of requirements was not directly met by any existing methods, so I began the development of the Eponine Anchored Sequence method, described below.

3.1. The Eponine Anchored Sequence (EAS) model

The Eponine Anchored Sequence model is a new approach to probabilistic sequence analysis which is capable of learning a complex overall model architecture and a set of small-scale sequence features in a single process. Mathematically, it can be represented as a generalized linear model, as described in the introduction to chapter 2.

EAS is a classification model which is designed to be applied to genomic contexts – that is, individual points within a large genome. In practice, genomic contexts are presented to the model as a large piece of sequence data and an integer defining the anchor point under consideration. When searching for features in bulk sequence, the same sequence is presented many times while scanning the anchor point along its length. The basic element of an EAS model is the positioned

constraint (hereafter, PC). This consists of:

- A preferred sequence motif, defined as a DNA position-weight matrix (page 1.4.1). Briefly, this is a list of columns, each defining one base of the motif, represented by a probability distribution over the DNA alphabet.
- A probability distribution over integer offsets relative to the anchor point, which defines the expected localization of the motif. In the work presented here, these distributions were always discretized Gaussians (*i.e.* the result of integrating the Gaussian probability density function over unit intervals). Gaussians were chosen because of their familiarity, and a smooth shape that made Gaussian-based models less prone to overfitting than functions with abrupt changes, such as square waves. However, any distribution over integers could in principle be used here.

To obtain a score for a PC on a given genomic context, the program scans over all positions in the sequence which are assigned a non-infinitesimal probability by the chosen position distribution. For each position, the probability of the sequence motif starting at that position being emitted by the chosen weight matrix is evaluated. The final score is given by:

$$\phi(C) = \frac{\log\left(\sum_{i=-\infty}^{\infty} P(i) \vec{W}(C, i)\right)}{|W|} \quad (3.1.1)$$

where C is a genomic context, P is a position distribution, and $\vec{W}(C, i)$ is a DNA weight matrix probability for offset i relative to the anchor point of C . Note the division by $|W|$, the number of columns in the weight matrix (*i.e.* the length of the sequence motif which it defines). This is important, since this method allow motifs with a wide range of lengths – with the trainer implementation described here, the length varies between 2 and 20 columns. The RVM trainer has a weak bias towards selecting basis function with higher absolute magnitudes. Normalizing the scores allows unbiased selection between motifs of different lengths. It is somewhat analogous to the whitening process often used to pre-process data for SVM classifiers, where all

the training vectors are normalized to constant length [Schölkopf *et al.* (eds.) 1999].

A single PC describes an individual sequence motif and its relationship to a point in a sequence, but a set of them can be combined to describe more complex structures. Figure 3.1 shows a schematic of a model combining three positioned constraints (note that in this schematic form, which is used throughout this chapter, the weight matrices are represented by single consensus sequences, which show the most likely symbol at each position in the motif). If the final output score is defined as a weighted sum of individual PC scores, the combined model is a generalized linear model over genomic contexts, with the PCs as basis functions. Therefore, it is possible use the sparse Bayesian learning methods from chapter 2 to reduce a large set of candidate PCs down to a sparse model containing a small, informative subset.

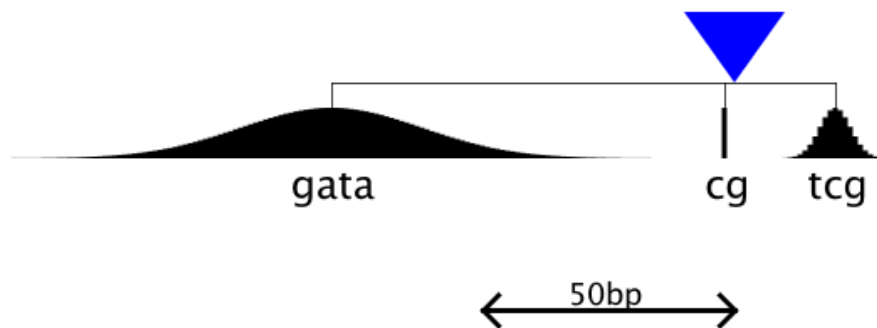


Figure 3.1. Example schematic architecture of an Eponine Anchored Sequence model.

There are several points to note about this approach to sequence modeling:

- While this description of the model architecture emphasizes detection of single, well-defined “words” in the sequence, since the overall PC score is based on a sum of weight matrix scores across a region, it is also possible to represent general compositional biases of a region by picking a PC with a short weight matrix and a very broad position distribution.
- It is quite acceptable for the distributions of two or more PCs to overlap. Since arbitrary weights are assigned in the training process to fit the model to the training data, any issues

with double-counting of a particular piece of information are corrected automatically

- While in this work, the models consist purely of PCs of the form defined above, it is possible to include completely different forms of basis function in the model. Once again, the training process will weight different types of evidence appropriately.

One limitation of the basic EAS model is that it is not able to capture interactions between pairs of motifs. Consider two motifs: A is found in the range [50:100] relative to an anchor point, while B is found in the range [30:80]. However, the spacing between the two motifs is much more conserved: always in the range [18:22]. An EAS model can capture the two motifs, and their broad position distributions, but misses the additional information in the covariance of the two motifs' positions. Including this information while maintaining the useful property of each motif being a single basis function would break the restrictions on generalized linear models, and therefore prohibit the use of the closed-form training algorithm from chapter 2. A possible solution would be to use more complex basis functions, each of which represented a small “scaffold” of several motifs, with particular spacings between them. This makes the space of possible basis functions far more complex. This makes the training procedure substantially more complex, and was not found to be helpful for the problem considered. Scaffolds *are* used in the alternative model described in chapter 4.

3.1.1. Learning EAS models

The space of potentially interesting PCs is extremely large. Even taking a highly simplified view, restricting the constraints to simple motifs rather than weight matrices and the position distributions to Gaussians of a constant width, there are over one million PCs representing six-base motifs with distributions centred at positions in the range [-250:50] relative to the anchor point. This space is too large to search exhaustively. Of course, in practice the EAS framework allows for an infinite (limited only by numerical precision on the probability values) number of PCs. Fortunately, as in the case of radial basis functions discussed in the chapter 2,

the space of possible basis functions is highly correlated. Making a small change to, say, one of the probabilities in a weight matrix will give a second PC whose output on a given sequence is correlated with the first. For this reason, exploring regions of a conceptual “PC-space” in the neighborhood of constraints which have already proved to be informative is likely to reveal even more informative constraints.

Taking advantage of these correlations, EAS models were trained using the two-class sparse Bayesian GLM trainer described in chapter 2, with a sampling strategy to create new basis functions. When the size of the working set fell below the low water mark, the trainer selected sampling strategies at random from the following set:

- Constructing a new PC, not based on the current set. This is performed by the following algorithm:
 - i. First, select a context at random from the training set (either positive or negative, without bias).
 - ii. Pick some point relative to that context’s anchor point.
 - iii. From that point, take a sequence motif of between 3 and 6 bases in length, and construct a weight matrix which optimally matches that consensus sequence, but includes some degree of uncertainty.
 - iv. Construct a PC using the newly selected weight matrix with a Gaussian position distribution of random width, centred at the position at which the motif was originally found.

Obviously, PCs selected in this way will strongly match the training example from which they were originally derived. This is closely analogous to selecting radial basis functions centred on points in the training data set, as used in the examples in chapter 2.

- Selecting an existing PC and adjusting the emission spectrum of one column of its

weight matrix, by sampling from a tightly-focused Dirichlet distribution centered on the current values.

- Adding an extra column to either the start or the end of an existing weight matrix, up to a maximum number of columns (in this case 20).
- Removing the start or end column from an existing weight matrix, down to a minimum of two columns
- Adjusting the width parameter of a Gaussian position distribution
- Adjust the centre position for a Gaussian position distribution

For an initial period of 200 cycles, only the first sampling rule (creation of novel basis functions) was used. After this point, the full range of sampling strategies were available, and training proceeded by a mixture of sampling and introduction of novel PCs.

3.1.2. Implementation and validation of EAS

The EAS model system, and the various sampling rules described above, were implemented running on the Java 2 platform, using components from the BioJava library [The BioJava development group, <http://www.biojava.org/>] to load and manipulate sequence data and probability distributions. Training was performed using the Java variational RVM training library described in chapter 2.

Before commencing work on real datasets, I wished to validate the training mechanism to ensure that it could build viable classifiers, and that it would correctly recover known information from the training set. Therefore I constructed a synthetic dataset consisting of unbiased random sequence (i.e. a list of samples from a uniform distribution over the alphabet of DNA symbols) into which were inserted three motifs: “GCAAT”, “TAGGAT”, and “ACGTAC” with some variability (for each motif-instance, either zero, one, or two bases were changed relative to the consensus pattern), and some variability (5 bases in either direction) in the position

with respect to the anchor point. Clearly, this dataset provides a good target for modeling with the EAS framework, since its construction closely matches the principles which were assumed when designing the model. Therefore, this test is simply a validation of the implementation and training method, rather than confirming that EAS will be able to answer real biological questions. The training data consisted of 100 of these synthetic sequences as positive examples, and 100 unspiked unbiased random sequences as negative examples.

Models were trained for 3000 cycles using the VRVM training module described in chapter 2, in small working set model with a high water mark of 27 basis functions and a low water mark of 24. Monitoring of the trainer showed that the working set reached low water mark and was topped up every 5 to 10 cycles of training, with no cases of ‘stuck’ training (where no further basis functions can be removed). Checkpoints of the trainer state were stored every 100 cycles for later evaluation. The final model is shown in figure 3.2. After 3000 cycles, the three motifs used to build the synthetic dataset were recovered perfectly. This was generally quite reproducible, although in some cases the first or last base of one of the motifs would be missed. However, recovery of information from this training set was generally excellent, indicating that the RVM-based training approach can be applied to sequence data, and that the pragmatic sampling strategy is able to successfully train models of this complexity.

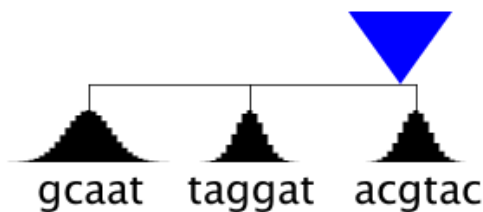


Figure 3.2. Schematic of an EAS model learned from the synthetic dataset, showing the three spiked motifs.

To monitor the progress of the training, and also to verify that overtraining did not reduce the predictive power of the model, I tested the checkpoint models produced while training the model in figure 3.2. Figure 3.3 shows two learning curves indicating the proportion of data which was correctly classified by the various checkpoint models. In the first curve, a threshold

probability of 0.5 is used – in other words, all sequences are assigned to either the positive or the negative class. By the end of the training process, almost all of the data is assigned to the correct class by this criterion. The second curve only counts positive test sequences as correctly classified if the model $P(\text{positive}) > 0.99$, and similarly $P(\text{positive}) < 0.01$ for negatives. Any example with an intermediate model output is counted as unclassified. Early in the training process, few sequences receive such a high-confidence classification, but after 1000 cycles, around 80% of the data is correctly classified with this level of confidence. The proportion increases only very slightly after this. The sequences which are never classified with high-confidence predictions are mainly examples from the positive class with large numbers of mismatches in the spiked motifs. While both learning curves fluctuate slightly, there do not appear any substantial increases in the proportion not correctly classified. This suggests that overtraining is not a serious problem with this type of model.

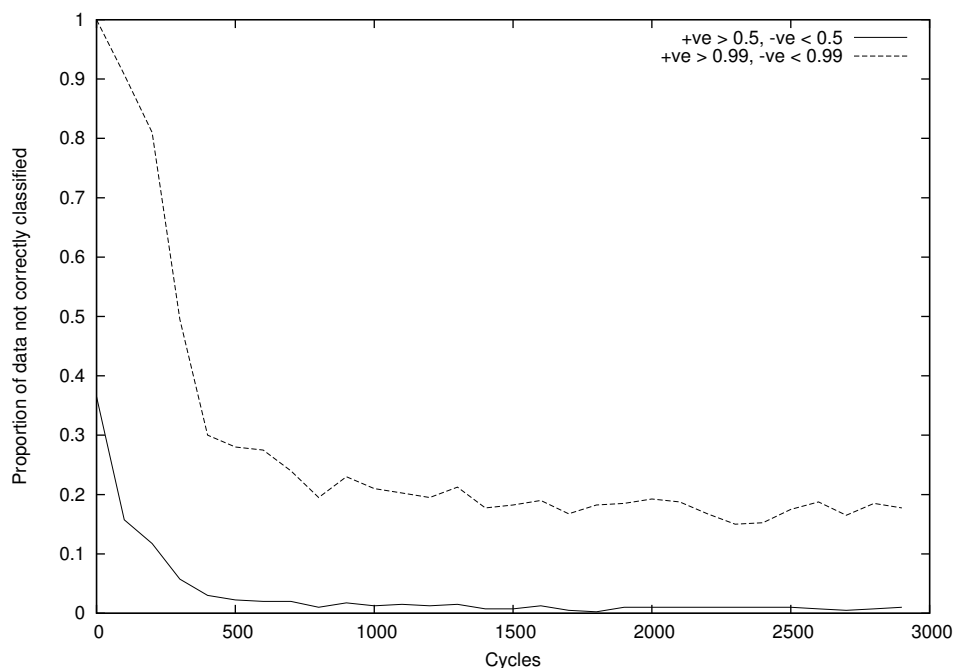


Figure 3.3. Learning curves for the training of an EAS model on the synthetic dataset.

Finally, I tested the predictive power of the learned model by plotting a receiver operating characteristic (accuracy vs. coverage) curve, shown in figure 3.4. Accuracy remains at 100% up to a coverage of 94% showing that, with a suitably chosen threshold, the model is a powerful

predictive tool when working with this kind of data.

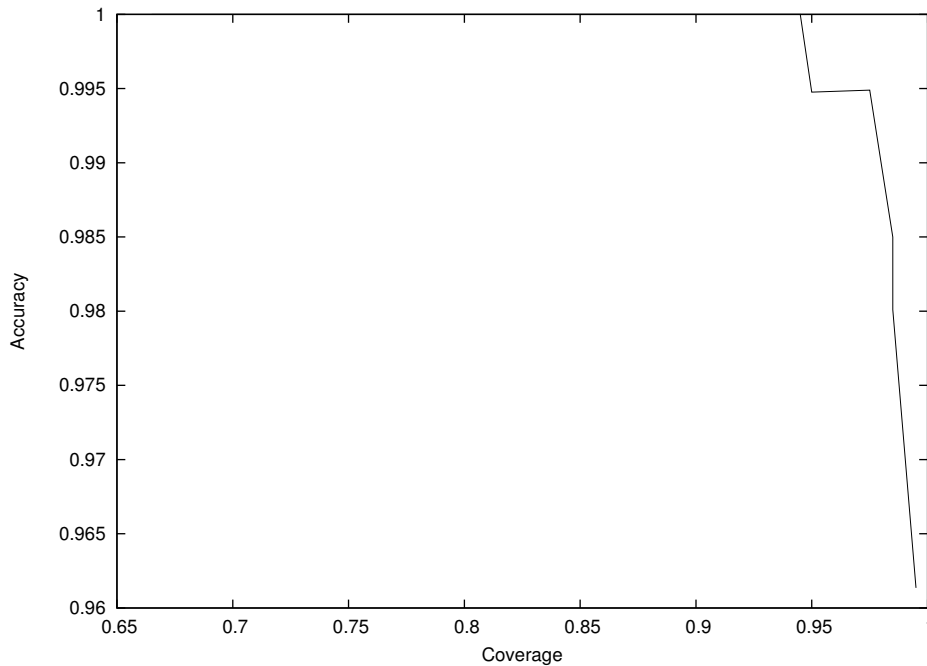


Figure 3.4. Accuracy vs. coverage (ROC) curve for a model trained on the synthetic dataset.

3.2. Training a transcription start site model

As discussed in chapter 1, sources of data about the precise location of vertebrate transcription start sites are relatively limited. For initial training, 389 mammalian promoters were extracted from release 62 of the EPD database [P erier *et al.* 2000]. I used the standard EPD web tools to download a non-redundant (defined by EPD as <50% sequence identity) set of sequences, extracting sequence data in the interval [-499:100] relative to the experimentally-mapped transcription start sites. Of these sequences, 50 were held aside for later use as an independent test set. This left 339 sequences which were used for training purposes. Around half of the EPD sequences were of human origin, but none used in the training set could be mapped to human chromosome 22 – this is important since chromosome 22 sequence was used as test data for some of the evaluation procedures used later in this chapter.

The trainer also requires a set of examples which are representative of the negative (non-promoter) class. For this purpose, I used fragments selected at random from the final introns of multi-intron genes on human chromosome 20. Introns early in a gene may often contain alternate transcription start sites [Laurinn *et al* 2000], but it seems unlikely that this will be the case for final introns, making them a good choice as representative non-promoter sequences. All negative and positive sequences were of the same length, and I picked an equal number of negative and positive sequences.

Models were trained using the procedure described previously, except that training was continued for 5000 cycles. Training typically took less than three hours on a typical personal computer (300MHz Pentium II CPU). Once again, there were no cases where the training process got stuck. A typical model from this process was selected as the EponineTSS_1 model, and is shown in figure 3.5. Unlike the models trained from the synthetic data, the learned models contained PCs which were assigned negative weights in the generalized linear model. These are shown in blue in the schematic diagrams. The obvious interpretation of these is that the presence of a particular motif at a particular position actually makes it less likely that this is a promoter sequence. Whether these are genuine negative signals, or whether they are simply artifacts of the training process remains to be seen, although results later in the chapter appear to favor the latter possibility.

Inspecting this model, it can be seen that all learned PCs target sequences less than 150 bases upstream of the anchor point, despite the inclusion of 500 bases of upstream sequences in the training set: this model is targeting a rather compact area of sequence. This result also confirms that the sparsity properties of RVM learning can indeed be extended to the field of sequence analysis. There are two motifs with distribution means at -29 and -30 which are both A/T rich. I believe that these are related to the TATA box described in classical promoter literature [Bucher 1990]. These, or similar, motifs were detected at around this position in all models inspected, with the presence of two A/T-rich PCs with strongly overlapping distributions occurring very frequently. Most of the other PCs in the model are C/G-rich (note several

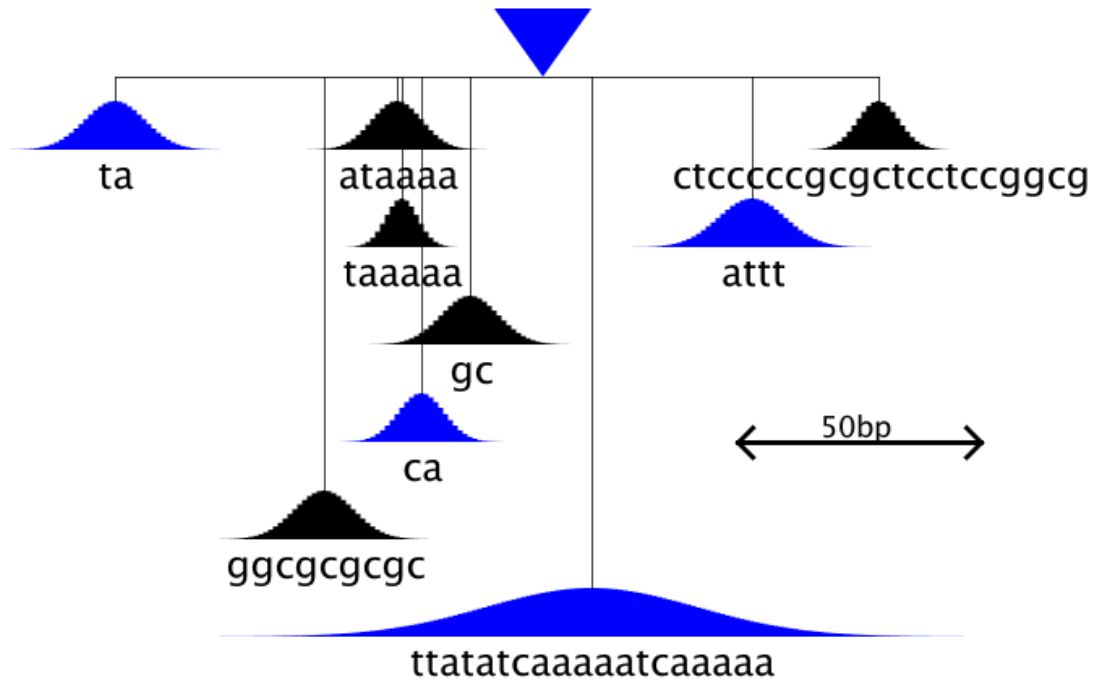


Figure 3.5. The EponineTSS_1 model, trained from 389 mammalian sequences from EPD.

A/T-rich motifs which have been assigned negative weights in the GLM, signaling a preference for “not A/T” in that region). However, the exact sequences and positions of these PCs varied substantially between training runs. I believe that this instability was due to lack of training data, or possibly the inclusion of some strongly atypical promoters in the positive set. In order to improve the quality of the learned models, I therefore decided to consider alternate sources of training data.

3.3. Model refinement using mouse cDNA sequence data

An alternative source of data, which became available during the course of this project, is the results from high throughput cap-trapped cDNA sequencing projects, such as the mouse FANTOM project [Kawai *et al.* 2001]. These projects use a molecular biology trick, outlined on page 16, to preferentially clone cDNA copies of full-length messenger RNA molecules. In principle, full-length cDNA sequencing seems an attractive way to discover transcription start

sites. However, there is still some uncertainty remaining as to whether cap-trapped cDNA clones really tell us about the true transcription start site, or if there is still a degree of truncation despite the cap-trap method. To demonstrate the causes for concern, I considered 68 human genes which had transcription start sites annotated in both EPD release 71 and DBTSS, a human full-length cDNA project based on similar technology to FANTOM. The sequences were mapped onto the current human genome assembly using the SSAHA fast sequence-matching package [Ning *et al.* 2001]. In many cases, the expected transcription start sites from the two methods differed significantly: a histogram of offsets is shown in figure 3.6. While the fact that differences can be seen is not entirely surprising – many cases of alternate transcription start sites have been observed in eukaryotes, and I believe that this is probably a common phenomenon – it is surprising, and a possible cause for concern, that the transcription start site according to DBTSS is more likely to be downstream of EPD than *vice versa*.

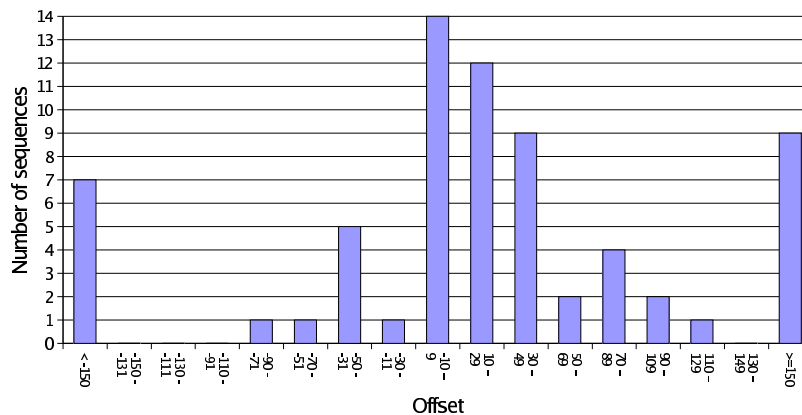


Figure 3.6. Histogram showing offsets of DBTSS start sites relative to those of corresponding EPD entries.

While the number of genes considered here is unfortunately rather too small to draw particularly strong conclusions, it appears that there seem to be approximately equal number of cases where DBTSS and EPD evidence for transcription start sites differs by a significant amount (>150 bases). These are likely to reflect *bona fide* alternate transcription start sites. However, when a small difference occurs, it appears that the DBTSS evidence is likely to be truncated. There are several plausible ways in which truncated mRNAs could be accidentally tagged by the cap-trapping process. Firstly, it is possible that the phosphatase treatment may not go

completion, leaving some capless mRNAs which retain their 5' phosphate groups. Second, it is possible that some RNA degradation could occur between the removal of the cap and the ligation of the tag – RNA is, after all, a rather unstable molecule and can undergo spontaneous hydrolysis. Despite these concerns, cap-trapped cDNAs still offer an interesting window onto the cell's transcriptome, but some care should be taken when using them as evidence of TSS position.

Simply sequencing cDNAs does not, by itself, give promoter sequences. However, in addition I had access to a repository of raw trace sequences [<http://trace.ensembl.org/>] which, at that time, contained around 1x coverage of the mouse genome, produced in the early phases of the the whole-genome shotgun sequencing effort [MGSC 2002]. Each trace was simply is simply the result of a single sequencing run, so the error rate is likely to be higher than that for assembled sequence. But modern sequencing methods are fairly accurate, and I considered trace data to be good enough for training this kind of model. To identify promoters in the trace repository, I searched 100 base fragments from the 5' ends of 19168 FANTOM cDNA sequences against the full set of sequences, again using SSAHA.

Using this procedure, I was able to retrieve trace sequences containing an exact match for 9958 cDNA ends. Of these, 3813 traces had at least 150 bases upstream of the mapped cDNA end. These were used as the basis for the second-round training set. But as shown above, it seems likely that many of these cDNAs may be slightly truncated. Therefore, I used the EponineTSS_1 model to scan the sequence 20 bases upstream of the mapping point for the cDNA 5' end, to filter this set and create a dataset whose entries are very likely to represent true transcription start sites. When an EponineTSS_1 prediction with a score of at least 0.999 occurred in this region, the sequence was accepted and the TSS annotation was adjusted to the point with the highest EponineTSS_1 score.

Finally, this process left a set of 599 mouse sequences with at least 130 bases of upstream sequence, anchored with a high degree of confidence at the true transcription start site. As before, a negative training set of equal size was built using final-intron sequences, and the

VRVM trainer was run using the same configuration as before. The learned model, known as EponineTSS_2, is shown in figure 3.7.

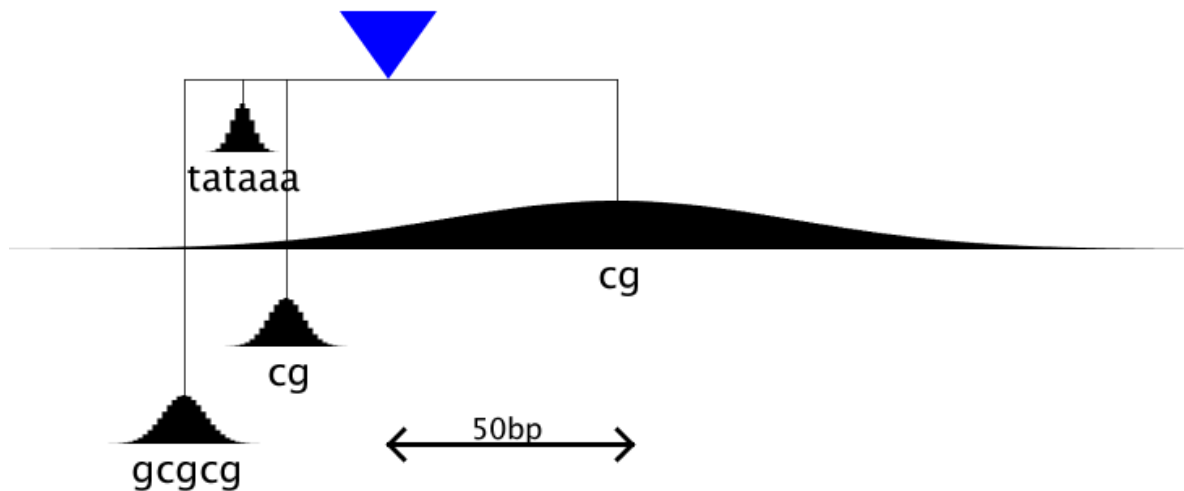


Figure 3.7. The EponineTSS_2 model, trained from 599 mouse sequences.

This model is dramatically simpler than the previous versions from figure 3.5. I note a single clear motif (position -30) which would appear to represent the TATA box. This is closely flanked by two C/G-rich motifs. Finally, there is a preference for C/G enrichment, primarily in the region immediately downstream of the transcription start site. I presume that this part of the model is detecting a signal related to the previously reported CpG islands in promoter regions – however, this model taken as a whole is clearly more than a straightforward CpG island detector. In this case, all learned PCs were given positive weights in the GLM. This suggests that the PCs with negative weights seen previously may have been artifacts, or reflect problems with the training data.

3.4. Validation and testing of EponineTSS

I was not able to identify a single, self-contained, dataset which tested every aspect of the EponineTSS model. The EPD database contained accurately mapped transcription start sites, but did not cover any single large region of genomic sequence. On the other hand, while

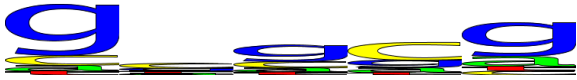



Motif	Centre	Weight
	-42	6.23
	-30	8.64
	-21	3.52
	47	12.3

Table 3.1. The four position weight matrices used in the EponineTSS_2 model.

curated annotation of finished human genomic sequence was underway, this did not include such accurate localization of transcription start sites. I therefore used a two-stage evaluation procedure to consider accuracy both in terms of detecting the exact position of transcription initiation and the rate of overprediction in bulk genomic sequence.

3.4.1. Testing on human chromosome 22: bulk genomic performance

Since genome sequencing is now a large-scale operation, it is important that analysis tools perform well across large stretches of sequence, rather than just small regions being subjected to individual detailed analysis. Therefore, I assessed the performance of the EponineTSS model on human chromosome 22. This was an early milestone in the human genome finishing process, and at the time of its publication was the largest continuous piece of finished sequence

[Dunham *et al.* 1999]. When I was developing the EponineTSS models, version 2.3 of the chromosome 22 annotation [Dunham, personal communication] was the best-annotated large region available. Unlike some other regions, where annotators have concentrated entirely on annotating the coding regions of genes, chromosome 22 includes many annotations of complete transcribed regions, including full UTRs, based on EST and cDNA evidence and experimentally validated using the Rapid Amplification of cDNA ends technique [Schramm *et al.* 2000]. While this still is not guaranteed to pinpoint the actual TSS, it at least increases the chance of finding the correct first exon in genes with interrupted 5' UTRs.

Unfortunately, not all genes were validated to an equal degree. Out of 618 annotated genes, 284 were marked “GD_mRNA”, indicating that they are presumed complete gene structures with supporting experimental evidence. I therefore believed that the true TSS is likely to lie close to the annotated 5' end of these structures. The remaining annotations are not necessarily complete, and I wished to treat them with a degree of caution. In particular, it is quite possible that a prediction made some distance upstream of one of these structures could be the true transcription start site, if the true first exon is missing from the gene structure. Therefore, I decided to construct a large synthetic DNA sequence – a pseudochromosome – consisting exclusively of those portions of chromosome 22 where the annotation which could be treated with reasonable confidence.

The pseudochromosome was constructed by selecting all regions containing GD_mRNA genes (including as much flanking sequence as possible), while rejecting regions containing partial and unverified genes (figure 3.8). The problematic case occurred when a GD_mRNA gene appeared adjacent to a partial gene in the opposite orientation, with both genes transcribed outwards from a common intergenic region: a pair of divergent genes. In most cases, these are assumed to be entirely separate transcription units, although there are thought to be some closely spaced cases where two genes are transcribed from a single regulatory region [Asakawa *et al.* 2001]. When preparing the pseudochromosome, the intergenic region of divergent pairs was split midway between the two gene structure annotations.

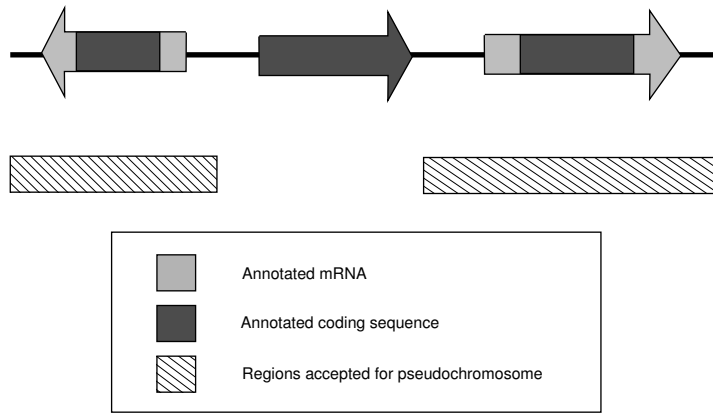


Figure 3.8. Selection of regions to include when the pseudochromosome sequence.

A pseudochromosome constructed in this way should be reasonably representative of chromosome 22 as a whole. This is, however, considered to be a relatively gene-dense chromosome by the standards of the human genome as a whole [Dunham *et al.* 1999], so some care should be taken when extrapolating results to other systems.

Both the EponineTSS_1 and EponineTSS_2 models were run across the pseudochromosome at a range of thresholds, from 0.97 (extremely low stringency) to 0.99999 (only a very small number of predictions). Most predictions fell into small clusters, with sizes generally in the range of 50-1000 bases. These clusters seem likely to reflect alternate transcription start sites of a single, but since there is some uncertainty in the actual TSSs of the chromosome 22 genes, and no annotation of alternate TSSs, it was not possible to directly validate this hypothesis. Instead, all predictions were subjected to single-linkage clustering, joining all predictions at a distance of 1000 bases or less. Around 60% of clusters included predictions on both positive and negative DNA strands, so the strandedness of predictions was ignored at this point – this is discussed later. I counted coverage as the proportion of genes which had a prediction cluster overlapping a window stretching 2kb upstream of the annotated start, and accuracy as the proportion of prediction clusters which overlapped one of these windows. Receiver operating characteristic curves for both models are shown in figure 3.9. Note that at thresholds giving coverages of less than about 0.1, the total number of predictions considered

was rather low, therefore the accuracy figures for the points on the extreme left of the curves should be treated as relatively poor estimates.

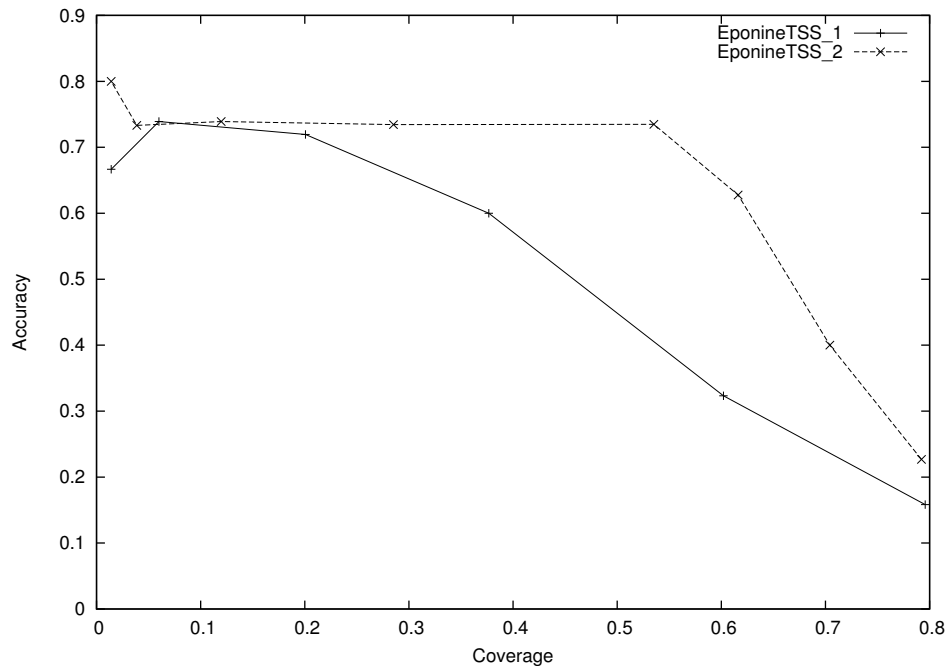


Figure 3.9. Accuracy vs. coverage for two EponineTSS models on the pseudochromosome.

From figure 3.9, it is clear that the EponineTSS_2 model is significantly and unambiguously the better of the two predictive models, in terms of giving a substantially higher accuracy for a given coverage across a wide range of thresholds. Moreover, the accuracy of the EponineTSS_2 model reaches a plateau of around 73% at a threshold of 0.999, and does not vary significantly from this when used at higher thresholds. The maximum accuracy of the EponineTSS_1 model is very similar, but is only achieved at a much lower coverage. I therefore chose to use the EponineTSS_2 model at a threshold of 0.999 as the final “product” of this approach.

While I do not consider this dataset to be an ideal test of positional accuracy, it remains interesting to see how well the predictions correspond to the annotated gene starts. Therefore, predictions from the EponineTSS model were placed into 50bp-wide bins according to their position relative to the annotated start of the nearest GD_mRNA gene, giving the density histogram in figure 3.10. While there is some variation in position, the vast majority of

predictions are within 1000 bases of the annotated start, and the distribution is sharply peaked around 0. In a large proportion of cases, the predictor and the curated annotation agree quite well.

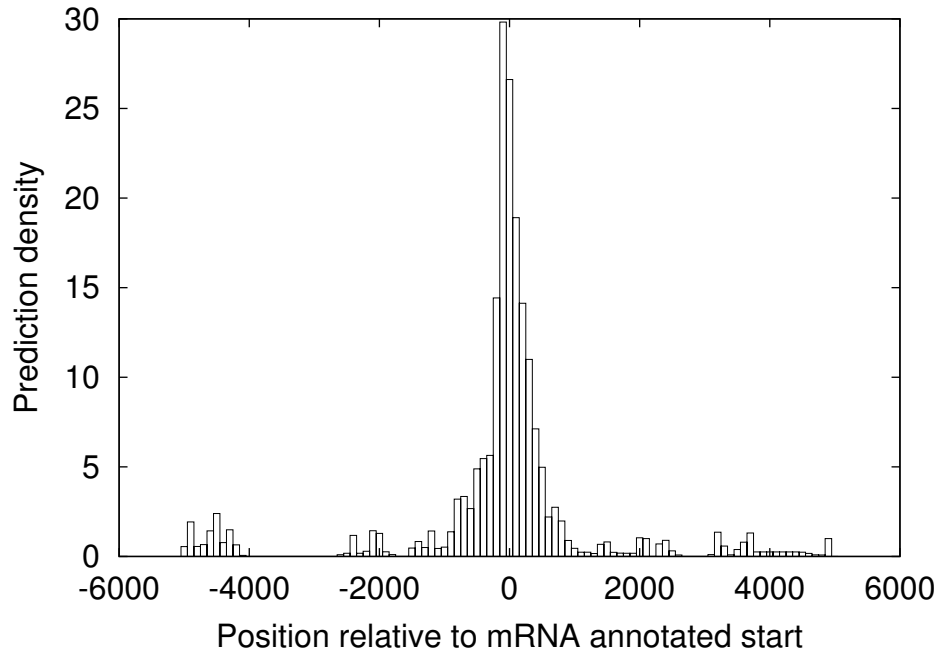


Figure 3.10. Density of prediction from the EponineTSS_2 model relative to annotated gene starts.

More recently, a new (version 3) assembly of chromosome 22 has been published, and I obtained an updated version of the gene annotation, described as version 3.1b [Dunham *et al.*, personal communication]. This included more GD_mRNA confirmed gene structures, and also improved annotation of pseudogenes and non-coding RNA genes. A new pseudochromosome was prepared using the same protocol as before, excluding all genes except those designated GD_mRNA. The resulting sequence was similar to the original pseudochromosome, but was slightly longer (19.3Mb), and included an additional 76 GD_mRNA genes, bringing the total to 360. Running the EponineTSS_2 model with a threshold of 0.999 gave a coverage of 54.4% – not a significant difference from the 53.5% obtained with the earlier annotation – but accuracy had increased slightly from 73.5% to 77.6%. Plotting the full ROC curve (figure 3.11) shows that the accuracy was consistently a little higher over a wide range of coverage values. This suggests that at least some of the false positives are real transcription start sites which will no longer

appear as false positives as the standard of annotation improves.

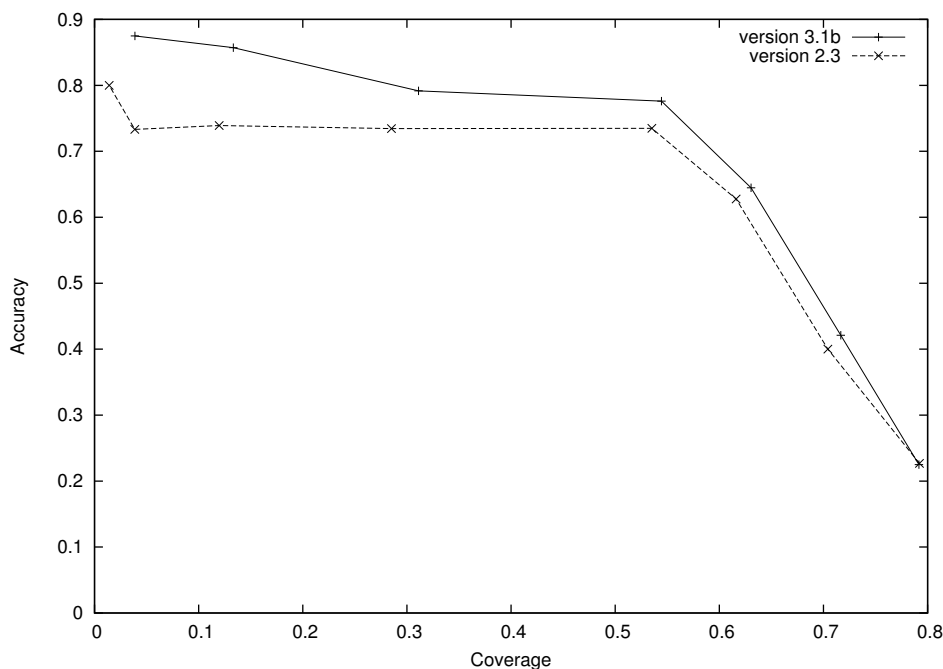


Figure 3.11. Accuracy vs. coverage for the EponineTSS_2 model on pseudochromosomes based on old (2.3) and new (3.1b) curated annotation of chromosome 22.

3.4.2. Testing on EPD: calibration of positional accuracy

The specifications for the EPD database require transcription start sites to be mapped experimentally to an accuracy of ± 5 bp, making it the most precise resource available for evaluating the positional accuracy of the model's predictions. I used the EponineTSS_2 model to scan the 50 EPD entries which were not used in the phase-one training process. A histogram of the prediction positions relative to mapped TSSs is shown in figure 3.12.

The results show a clear peak, with predictions clustered in the interval $[-10:20]$ relative to the EPD-specified transcription start site – only a little more than the ± 5 base pair tolerance specified for EPD mapping. Since EPD is a resource made up from data submitted or published by a large community of researchers, it is not known for certain if the ± 5 criterion is actually true. However, this result suggests that EponineTSS predictions are very likely to be within 10 bases of the true TSS. If EPD mapping is slightly less accurate than stated, it is possible that EponineTSS

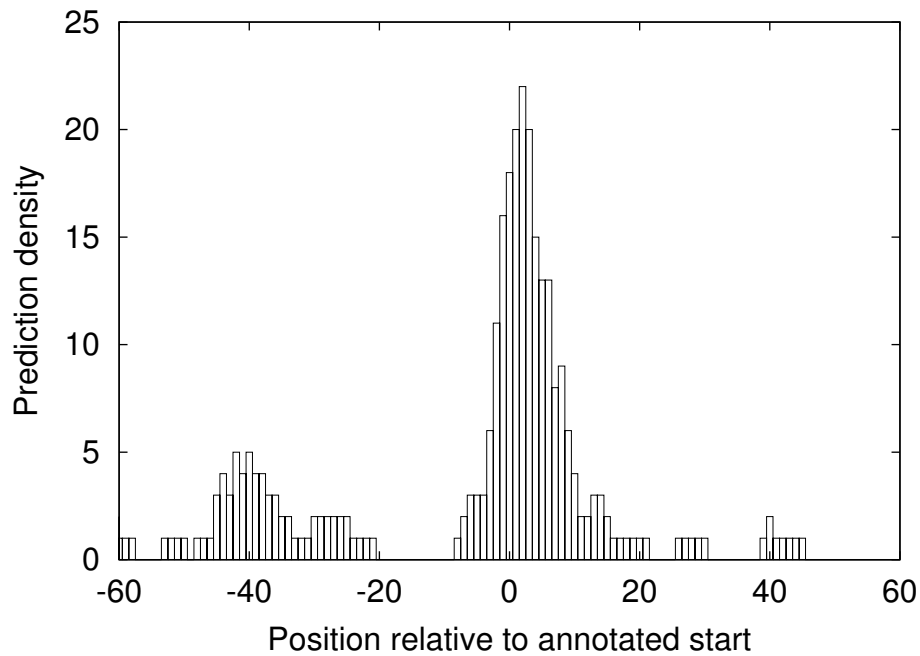


Figure 3.12. Density of EponineTSS_2 predictions relative to the annotated TSS of EPD entries.

accuracy may be higher than this. A second, much smaller, peak occurs around -40 relative to the true TSS: this might represent a common spacing for alternate transcription start sites, but it could also be noise in the results – especially since 50 sequences represents a relatively small test set.

3.4.3. Comparison with other methods

I compared EponineTSS specificity and coverage on the human pseudochromosome sequence. Three other methods were considered:

- A generic CpG island detection program, as described on page 22, which was run by the chromosome 22 analysis group [Dunham *et al.* 1999].
- The PromoterInspector program [Scherf *et al.* 2000], a dedicated method for prediction of promoter regions.
- A DNA weight matrix for the TATA box, derived by alignment of EPD sequences

[Bucher 1990].

Due to availability restrictions, I could not download PromoterInspector and directly scan the chromosome 22 sequence, and limitations on the web-based version of the program prevented scanning large pieces of sequence. However, the authors had previously published a set of predictions on chromosome 22 [Scherf *et al.* 2001], and I was able to extract the complete set of predictions from their web pages. Unfortunately, these predictions were published in full-chromosome coordinates for an earlier assembly of chromosome 22, while I was keen to use the more recent version 2 assembly, which had better gene annotation with many more GD_mRNA confirmed gene structures. Without the ability to rerun PromoterInspector, it was not possible to make a direct comparison. Therefore, I extracted DNA sequences for the region of each prediction from the original assembly and used SSAHA [Ning *et al.* 2001] to find perfect matches on the version 2 assembly. In this way, I mapped 99.4% of the original PromoterInspector predictions. Therefore I believe that results given here for PromoterInspector should be representative.

The TATA box weight matrix (shown graphically in figure 1.8) was downloaded from the EPD website [EPD, <http://www.epd.isb-sib.ch/>]. From this website, I also obtained a recommended log-odds score threshold of -6.5. However, scanning the full pseudochromosome with this threshold gave 39869 predictions – far more than any of the other methods considered here. Moreover, these were distributed quite uniformly across the chromosome and I was not able to reduce them to a more reasonable number by clustering. Therefore, I experimented with alternative thresholds, and found that a log-odds score of -2.6 gave 540 predictions, a number more in line with the other methods considered here.

Accuracy and coverage figures on the pseudochromosome were assessed as before, and results for all four methods are shown in table 3.2. I note that, at either of the thresholds tested, the accuracy of the TATA weight matrix is extremely low – this is clearly not an acceptable method for finding promoters in a genomic context

Method	Predictions	True positives	False positives	Coverage (%)	Accuracy (%)
Eponine	215	152	57	53.5	73.5
Pro'spector	278	157	100	55.3	64.0
CpG	306	187	116	65.8	62.1
TATA -2.6	540	37	500	13.0	7.4
TATA -6.5	39869	283	37581	99.6	5.7

Table 3.2. Sensitivity and selectivity of various promoter-prediction mechanisms on the human pseudochromosome.

The three remaining methods all offer much higher levels of accuracy, indicating that they can distinguish promoters from bulk genomic DNA. The coverage of all three methods is relatively similar. Moreover, the sets of genes detected by these methods are strongly overlapping, as shown in figure 3.13. This seems significant, since the three methods under consideration are technically quite different (although EponineTSS and PromoterInspector both used the EPD database during training). When this is taken in the context of the prior observation that further increasing the coverage for the EponineTSS_2 model means a severe loss of accuracy (figure 3.9, I believe that this indicates some significant difference between promoters which are detected by the methods considered here and those which are not.

3.5. Analysis of cases where promoters were not detected by EponineTSS

At least in this test region, the EponineTSS model was able to detect just over 50% of promoters. Noting that this set seemed to be largely common with other *ab initio* promoter-prediction method, I suspected that promoters could be subdivided into several classes, only one of which was being detected here – and which was also correlated with the previously noted phenomenon of CpG islands. I was therefore interested to see if I could learn anything else about this subset of genes, and also about the promoters which cannot currently be detected.

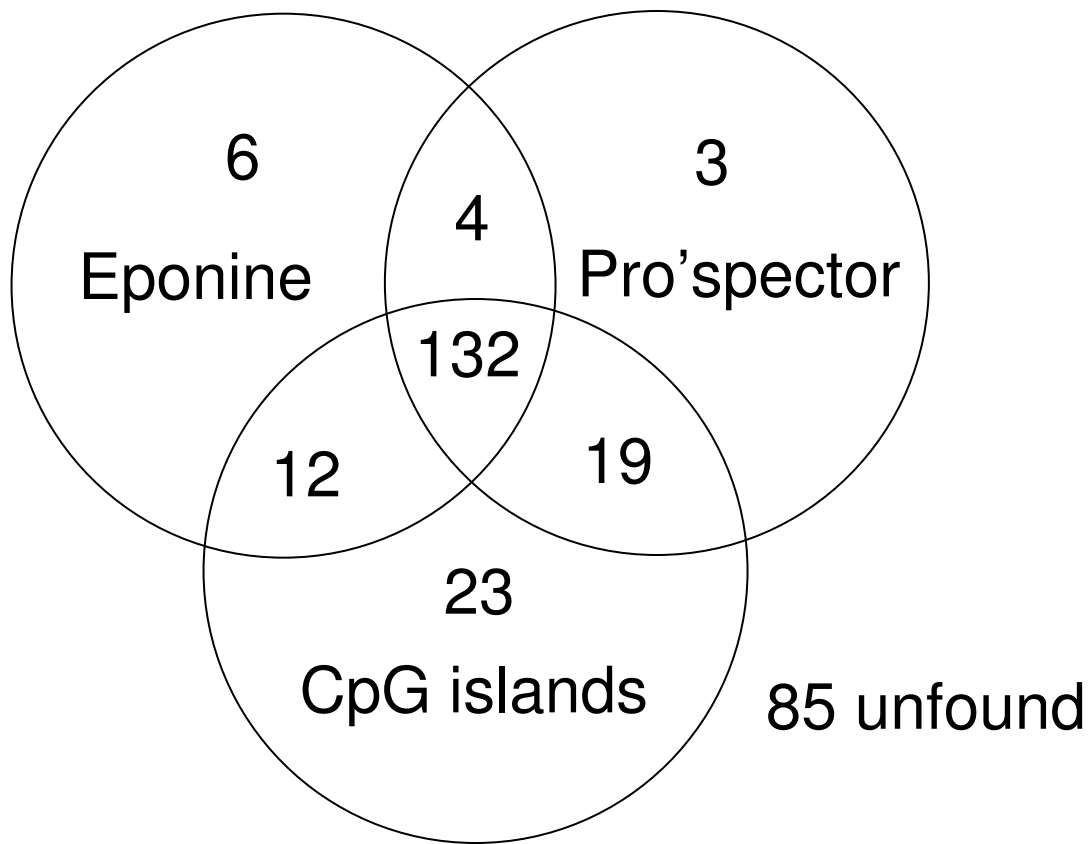


Figure 3.13. Intersection of “correct” predictions of promoters by EponineTSS, PromoterInspector, and CpG islands.

3.5.1. Modeling of non-detectable promoters

An obvious first step in analyzing the non-detectable promoters is to attempt to model them using the same approach as previously. At the time, the best candidate training set for this task was the set of mouse promoters previously derived from the FANTOM cDNA set. While previously, the EponineTSS_1 model was used to positively select a subset of these with detectable promoters, here a model was trained on sequences which do not receive any prediction. Unfortunately, in this case there was no obvious approach to align the training sequence at the transcription start site, so any model learned was unlikely to give the same positional accuracy as those considered so far. Once again, human final intron sequences were used as a negative set. When this process was carried out, a non-empty model was learned, suggesting that there *is* some information in this training set. However, the models are rather

complex, and had few consistent features between training runs. An example is shown in figure 3.14.

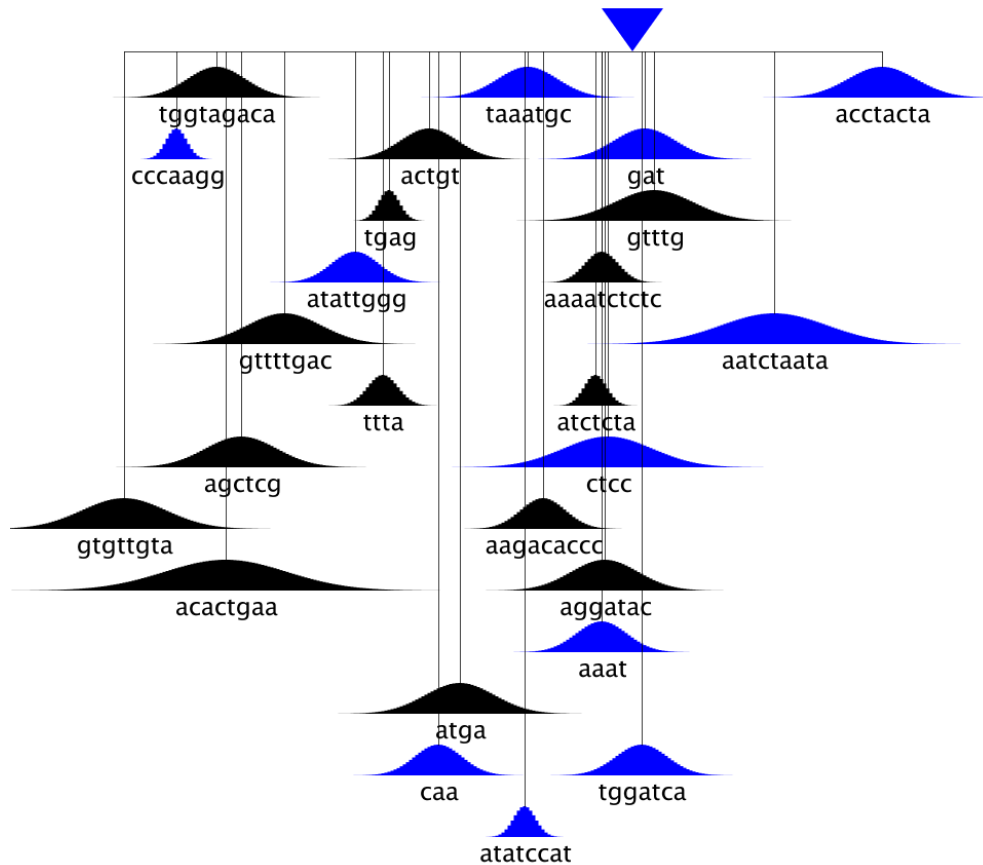


Figure 3.14. Model trained on negative-selected FANTOM data.

This model was tested on the pseudochromosome, using the same approach as before: see accuracy vs. coverage in figure 3.15. Regardless of threshold, this model gives a far lower accuracy than those considered previously – certainly too low to be a useful predictive tool on a genomic scale. To determine whether the model has captured any information about promoters at all, I compared it with the results from an entirely random predictor (i.e. assigning each point on the pseudochromosome a random score sampled from a uniform distribution over the interval [0.0:1.0]). This is the second trace in figure 3.15. This comparison shows that the negatively selected model does capture some information, but has very low predictive power compared to the previously described models.

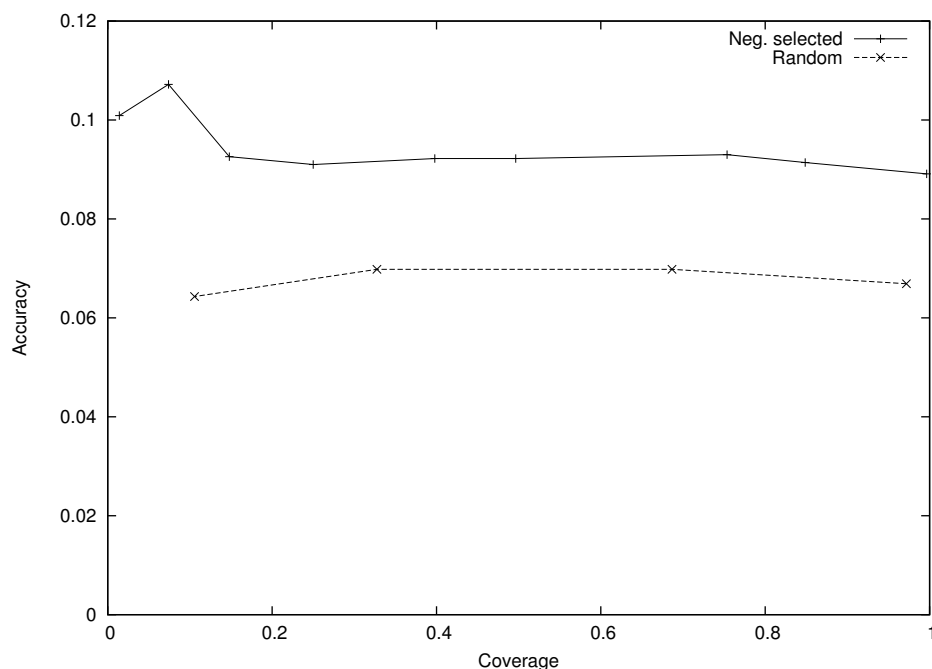


Figure 3.15. Accuracy vs. coverage for model trained on negative-selected FANTOM data, with random predictions for comparison.

I attempted to amplify the information contained in this preliminary model by using it to positively select a subset from the FANTOM dataset. However, the second-stage models produced in this way were not substantially simpler than the first-stage model, nor did they give a better prediction accuracy.

3.5.2. Correlation of promoter-detectability with gene type and function

I wished to determine if the distribution of promoters detected by the EponineTSS model was entirely random, or whether some classes of genes were much more likely to be regulated by detectable promoters than others. This question is particularly significant in the light of previous reports that CpG islands are associated primarily with “housekeeping” genes [Larsen *et al* 1992].

I concluded that the set of 284 chromosome 22 genes was too small to get a reliable impression of any correlation. Therefore, the complete set of Ensembl gene predictions for the

NCBI_30 assembly of the human genome was used – a total of 27,628 predicted transcripts from 22,980 predicted genes. The EponineTSS_2 model was used with a threshold of 0.999 to scan the entire genome. Transcripts were classified as:

- **Found** if a TSS prediction occurred within 2000 bases of the predicted gene start. This set consisted of 3765 transcripts.
- **Uncertain** if a TSS prediction occurred within 25,000 bases, but not within 2000 bases. This set consisted of 10948 transcripts.
- **Unfound** if no predictions occur within 25,000 bases. This remaining set consisted of 12915 transcripts.

The existence of the **uncertain** category aims to cover at least the majority of cases where the first (perhaps entirely non-coding) exon is missing from the gene prediction. Here, the uncertain cases are ignored and the emphasis is on comparing the found and unfound categories. It seems very likely that in many cases the predictions for the **uncertain** transcripts were, in fact, correct. However, the **found** category was sufficiently large that I did not consider it necessary to count the **uncertain** cases.

I wished to avoid lexical processing of gene description strings if possible. Instead, I relied on terms from the controlled vocabularies for molecular functions and biological processes provided by the GO project [The Gene Ontology Consortium 2000]. Automatic GO annotation is available for the bulk of Ensembl predicted transcripts, *via*. the GOA annotation of the Swissprot database [Camon *et al.* 2003].

I counted the proportion of both the found and the unfound transcript sets which was covered by each term in the GO ontologies. As well as counting terms applied by the GOA annotation, I counted all ancestor terms through the *is-a* and *part-of* relationships recorded in the GO database. For example, a gene annotated with the term GO:0003700 (“transcription factor”), as shown in figure 3.16 would also be counted as “transcription regulator”, “DNA binding”,

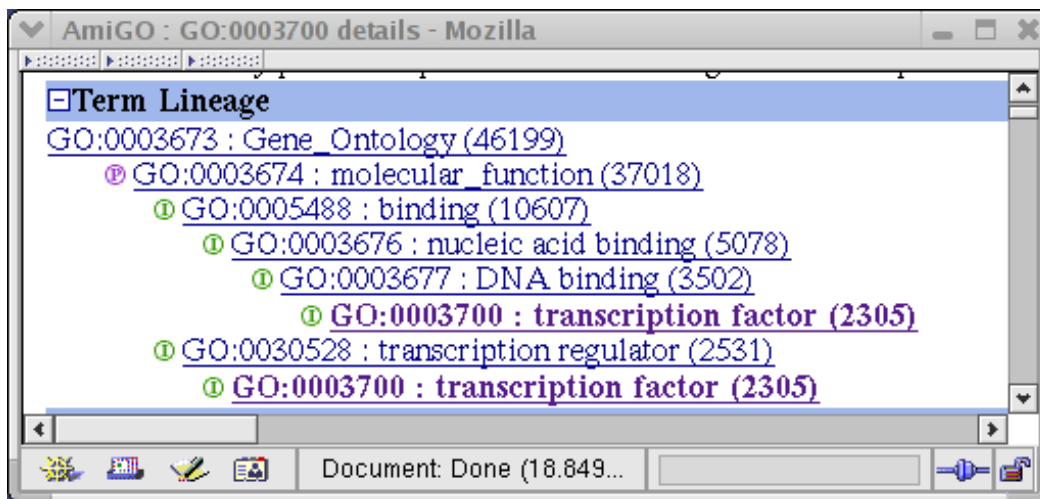


Figure 3.16. Example GO term lineage.

“nucleic acid binding”, and “binding”. Counting all levels of the ontology in this way avoids questions about which levels of the ontology would be most appropriate for comparing the two sets of genes.

This process resulted in a list of 3746 GO terms. These were ranked by the difference between the proportions of found and unfound transcripts labeled with the term (see tables 3.3 and 3.4). This means that a term which labels around 10% of both the found and unfound sets will appear near the middle of the listing. But conversely, a term which labels one found gene and no unfound genes will also appear close to the middle – a useful property, since a difference of a single gene is not statistically significant. The two tables show the sets of terms with the largest positive and negative differences respectively, and show terms which exhibit a clear and indisputable correlation with EponineTSS detectability. In the **found** column, a frequency of 0.01 corresponds to an actual count of 38 transcripts, while in the **unfound** column, a frequency of 0.01 means 129 transcripts. So most of the entries shown here are based on hundreds of transcripts, and are clearly significant.

Looking first at the list of terms overrepresented in the **found** category (i.e. large positive difference), near the top of the list we find “cell growth and/or maintenance”, “metabolism”, and “enzyme”. These are classically considered to be housekeeping functions, and seem to support

GO term name	Freq. (unfound)	Freq. (found)	Diff.
cell growth and/or maintenance	0.581	0.674	0.092
metabolism	0.400	0.488	0.087
ligand binding or carrier	0.439	0.507	0.067
DNA binding	0.118	0.176	0.057
enzyme	0.307	0.360	0.052
nucleic acid binding	0.193	0.243	0.050
nucleotide binding	0.101	0.148	0.046
purine nucleotide binding	0.101	0.147	0.046
nucleobase, nucleoside, nucleotide and nucleic acid metabolism	0.152	0.192	0.040
transferase	0.103	0.141	0.038
transcription	0.089	0.126	0.036
transcription regulator	0.052	0.088	0.036
transcription, DNA-dependent	0.088	0.123	0.035
transcription factor	0.046	0.080	0.034
protein metabolism and modification	0.147	0.182	0.034
protein modification	0.063	0.096	0.032
transcription regulation	0.083	0.115	0.032
kinase	0.049	0.081	0.032
phosphotransferase, alcohol group as acceptor	0.044	0.075	0.030
intracellular signaling cascade	0.065	0.091	0.026
transcription, from Pol II promoter	0.016	0.042	0.026
protein binding	0.070	0.096	0.025
protein kinase	0.035	0.061	0.025
ATP binding	0.078	0.103	0.024
cell organization and biogenesis	0.030	0.054	0.024
phosphate metabolism	0.046	0.070	0.024
adenyl nucleotide binding	0.079	0.103	0.023
guanyl nucleotide binding	0.022	0.046	0.023
protein serine/threonine kinase	0.022	0.045	0.022
developmental processes	0.086	0.106	0.020
cell cycle	0.046	0.065	0.019
GTP binding	0.017	0.036	0.018
transport	0.107	0.126	0.018
cell cycle control	0.014	0.031	0.017
transcription regulation, from Pol II promoter	0.008	0.025	0.017

Table 3.3. GO terms applied preferentially to **found** genes.

GO term name	Freq. (unfound)	Freq. (found)	Diff.
receptor	0.158	0.073	-0.085
response to external stimulus	0.135	0.058	-0.076
transmembrane receptor	0.122	0.046	-0.076
signal transducer	0.225	0.149	-0.075
defense response	0.090	0.025	-0.064
cell communication	0.325	0.265	-0.060
immune response	0.082	0.023	-0.058
response to biotic stimulus	0.095	0.037	-0.058
integral membrane protein	0.227	0.169	-0.058
G-protein coupled receptor	0.069	0.014	-0.055
rhodopsin-like receptor	0.060	0.009	-0.051
G-protein coupled receptor protein signaling pathway	0.081	0.033	-0.048
cell surface receptor linked signal transduction	0.105	0.061	-0.044
response to pest/pathogen/parasite	0.047	0.016	-0.031
defense/immunity protein	0.027	0.003	-0.024
perception of abiotic stimulus	0.032	0.009	-0.023
sensory perception	0.031	0.009	-0.022
integral plasma membrane protein	0.097	0.075	-0.021
response to wounding	0.028	0.007	-0.021
perception of external stimulus	0.043	0.022	-0.021
response to abiotic stimulus	0.042	0.021	-0.020
biological_process unknown	0.056	0.036	-0.019
RNA-directed DNA polymerase	0.018	3.969	-0.018
RNA dependent DNA replication	0.018	3.969	-0.017
cell adhesion	0.043	0.025	-0.017
signal transduction	0.206	0.188	-0.017
ligand	0.039	0.023	-0.016
stress response	0.055	0.039	-0.015
innate immune response	0.020	0.005	-0.014
perception of chemical substance	0.015	7.939	-0.014
chemosensory perception	0.015	7.939	-0.014
cytokine	0.023	0.008	-0.014
cellular_component unknown	0.057	0.042	-0.014
inflammatory response	0.019	0.005	-0.013
olfactory receptor	0.013	0.0	-0.013

Table 3.4. GO terms applied preferentially to **unfound** genes.

some connection between housekeeping and CpG-island-related promoters. However, the list also includes “transcription factor” and “protein kinase” – in fact, proportionately these terms are overrepresented by a much greater factor than “metabolism”. Yet expression of these genes is likely to be more restricted than that of metabolic enzymes.

Amongst the underrepresented terms, perhaps the most dramatic example is the “olfactory receptor” term, which makes up 1.3% of the unfound set, yet does not occur *at all* in the found set. Many other types of receptor also appear in this list. “G-protein coupled receptor” makes up 6.9% of the unfound set yet only 1.4% of the found set. Part of this is explained by the olfactory receptors (which are G-protein coupled), but other sub-types are also dramatically underrepresented. Other terms in this list include “immune response”.

The term showing the most dramatic difference – olfactory receptors – is perhaps the easiest to explain. The human genome is thought to contain well over 300 functional olfactory receptor genes [Crasto *et al* 2001], although not all of these are predicted by Ensembl. There are also many pseudogenes. But only one allele of one receptor locus is expressed in any particular olfactory neuron [Chess *et al* 1994]. The choice of receptor is presumed to remain constant throughout the neuron’s life. This seems to suggest that there is some specialized mechanism to select one particular olfactory receptor allele and promote the transcription of that – and no other – allele. Logic suggests that there might be advantages in using an atypical promoter system for these genes, since it lowers the probability of accidental transcription, compared to a common type of promoter sequence. Similar arguments could also apply to some components of the immune system. In addition, Ensembl identifies at least some of the fragments of the immunoglobulin light and heavy chains as genes, yet these would not be expected to have promoters at all, since they will only be expressed in cells where recombination events have occurred to create a complete immunoglobulin gene.

Some of the overrepresented terms are rather more surprising. Genes annotated as “metabolism” or “enzyme” fit well with the intuitive notion of housekeeping genes. Many

of these are required by most or all cell types for day-to-day survival. But the presence of transcription factors and other genes that play predominantly regulatory functions in this set does not fit this model and shows that the notion of a “housekeeping” gene is, at least, somewhat simplistic.

Another possible explanation for the difference is that gene products are required by the cell in radically different amounts. Many enzymes which catalyze key metabolic reactions are needed in significant amounts, simply to provide adequate throughput on a particular metabolic pathway. On the other hand a membrane-bound receptor can adequately perform its function with only a few copies present per cell. But once again, this fails to explain the presence of transcription factors on the overrepresented list, since they are not generally expected to be present in large quantities.

3.6. EponineTSS discussion

The results shown here demonstrate that a novel sequence analysis model, based on some previous suggestions of promoter structure, is able to capture information from a suitable training set. Representing the model within the GLM framework allowed a standard “learning engine”, which did not itself contain any domain-specific knowledge, to drive the training procedure, selecting and weighting the final set of model elements. While the emphasis here has been on modeling the sequence around transcription start sites, similar approaches might be suitable for other point features in the genome, and indeed the code is currently being tested for the prediction of transcription termination sites [A. Ramadass, personal communication].

The EponineTSS model is able to predict a substantial proportion of transcription start sites (over 50% of the pseudochromosome test set) with a comparatively low level of apparent overprediction (over 70% accuracy on a representative subset of human chromosome 22). This combines better accuracy than existing methods with a comparable level of coverage. The

EponineTSS scanner is fast and efficient, making it a convenient method for first-pass promoter annotation on bulk genomic sequence – and as such, it has been integrated into the Ensembl project's standard vertebrate analysis pipeline.

I find it striking that accuracy on the original pseudochromosome seems to peak at 73%, and neither re-trained versions of the EponineTSS model, nor any other method, were able to substantially exceed this. I believe that as our understanding of the genome improves, at least some of the false-positive prediction clusters will turn out to correspond with real promoters. Certainly, the improvement in accuracy when evaluating against the new version 3.1b chromosome 22 annotation seems to support this idea. On the other hand, it seems unlikely that all the remaining false positives correspond to protein coding genes: one possible explanation would be these promoters drive the expression of functional but noncoding RNA genes, such as the regulatory micro-RNAs [Grosshans and Slack, 2002], and it will be very interesting to watch as methods are developed which can more sensitively detect noncoding RNA genes, to see if some of EponineTSS' false positives turn out to be correct predictions.

Similarly, I note that this method, and many others, fail to predict more than 50-60% of promoters, at least without greatly increased rates of false-positive predictions, and that all current *ab initio* promoter-finding methods seem to detect similar subsets of promoters. I can show that the promoters correctly detected by this method are associated with a rather biased selection of genes. Olfactory receptor genes, whose expression is confined to small sub-populations of olfactory neurons, never have detectable promoters. This bias seems to extend to other types of receptor genes (I presume that many of these also have rather restricted expression patterns), and to components of the immune system. At the other extreme, enzymes are more likely than average to have a detectable promoter. This fits well with the existing suggestion that housekeeping genes have CpG-island promoters. But the same results also show detectable promoters associated with some of the key regulatory mechanisms of molecular biology: protein kinases and transcription factors. I suggest that, while most of these are not ubiquitously expressed, they are likely to be more widely expressed, in general, than the receptor

genes. Firstly, gene regulation is largely combinatorial, and it is a collection of transcription factors, rather than one single switch, which promotes gene expression. Secondly, regulators and signal-transducers are often re-used for multiple tasks in different cell types – for example, the MAP-kinase cascade [Nishida and Gotoh, 1993]. So while the idea that standard promoters are associated with housekeeping genes would certainly seem to be an oversimplification, it may well be true that promoters which include some standard core elements – and are detected by the methods considered here – tend to have more widespread expression patterns than those which do not.

One explanation for this is that the sets of training data used here (EPD and FANTOM) are significantly biased towards particular types of genes. This is an interesting possibility since PromoterInspector was also trained from EPD and exhibits the same bias as EponineTSS. There is more discussion of this in chapter 4. Assuming that the training set is not biased, another possibility is that there is some kind of functional bias in the correctness of Ensembl gene predictions, with certain types of gene more likely to have complete or near-complete UTR annotations, which are consequently more likely to match the TSS predictions. Alternatively, some types of gene may be particularly prone to duplication as pseudogenes, some of which are being predicted as false-positives by Ensembl but which lack functional promoters. Both of these hypotheses will become easier to test once curated annotation is available for a large fraction of the genome. The final possibility is that there is a genuine distinction between a “detectable” class of promoters, located by EponineTSS and PromoterInspector and generally associated with CpG islands, and the remaining promoters, which look quite different in sequence terms.

During the development of this method, I briefly experimented with two extensions to the basic EAS model. The first variant allowed PCs containing ‘scaffolds’ of two or more weight matrices, and was similar in implementation to the scaffold-based models in chapter 4. A second variant replaced simple position weight matrices with first order weight matrices, where the emission distribution at each position was conditional upon the symbol observed in the previous

position. This allowed dependencies between neighbouring bases to be modeled, as discussed on page 19. Both these variants allowed models to capture substantially more information than the basic EAS model, but did not give better performance for this particular task. This does not, however, rule out the possibility that they might be useful for some other sequence-analysis tasks.

So far, I have not been able to train a model to effectively detect the remainder of promoters. This may be due to limitations in the available training data (in particular, the fact that I could not precisely localize the true transcription start sites for the FANTOM promoters). However, it is also probable that the group of hard-to-detect promoters might be subdivided into several distinct types, without any strong common features. This could explain the complexity of the model in figure 3.14: the model includes elements from a number of unrelated different promoter types. In the future, this question could be addressed by taking a mixture-modeling approach which builds separate models for clusters of data in the training set, rather than forcibly fitting it to a single model.

One additional limitation is that the current models do not seem to effectively predict the direction of transcription from a given promoter, since the majority of prediction clusters include predictions on both strands. In some cases, such clusters can be explained by considering divergent genes, transcribed outwards from a single compact promoter region. A number of cases like this have been described, primarily in bacteria but also in eukaryotes, but there certainly are not enough divergent protein-coding genes to explain the large number of bidirectional clusters. One explanation would be that the “core” promoter signal seen here really does not provide substantial information about the direction of transcription, and that directionality is conferred by additional signals which are specific to individual promoters rather than being shared among large numbers of different promoters. Another, more radical, view is that divergent promoters such as that described in [Asakawa *et al.* 2001] – cases where a pair of closely-spaced genes are transcribed outwards from a single regulatory region – are more common than has so far been realized. While it is very unlikely that a large number of

genes have a so-far undiscovered protein-coding partner, it is much more plausible that there are many additional regulatory micro-RNAs, which could form divergent pairs with coding genes. Once again, until micro-RNAs can be accurately predicted, or experimentally detected in a high-throughput fashion, this will have to remain a tentative suggestion. In the mean time, it may still be useful to consider *EponineTSS* results when searching for novel RNA genes.