BMC Bioinformatics

Methodology article

Open Access

# Parameter estimation for robust HMM analysis of ChIP-chip data

Peter Humburg*[1,2], David Bulger[1] and Glenn Stone[2]

Address: [1]Department of Statistics, Macquarie University, North Ryde, NSW 2109, Australia and [2]CSIRO Mathematical and Information Sciences, North Ryde, NSW 2113, Australia

Email: Peter Humburg* - peter.humburg@csiro.au; David Bulger - dbulger@efs.mq.edu.au; Glenn Stone - glenn.stone@csiro.au

* Corresponding author

Published: 18 August 2008

*BMC Bioinformatics* 2008, **9**:343 doi:10.1186/1471-2105-9-343

Received: 30 May 2008

Accepted: 18 August 2008

This article is available from: http://www.biomedcentral.com/1471-2105/9/343



## Abstract

**Background:** Tiling arrays are an important tool for the study of transcriptional activity, protein-DNA interactions and chromatin structure on a genome-wide scale at high resolution. Although hidden Markov models have been used successfully to analyse tiling array data, parameter estimation for these models is typically *ad hoc*. Especially in the context of ChIP-chip experiments, no standard procedures exist to obtain parameter estimates from the data. Common methods for the calculation of maximum likelihood estimates such as the Baum-Welch algorithm or Viterbi training are rarely applied in the context of tiling array analysis.

**Results:** Here we develop a hidden Markov model for the analysis of chromatin structure ChIP-chip tiling array data, using *t* emission distributions to increase robustness towards outliers. Maximum likelihood estimates are used for all model parameters. Two different approaches to parameter estimation are investigated and combined into an efficient procedure.

**Conclusion:** We illustrate an efficient parameter estimation procedure that can be used for HMM based methods in general and leads to a clear increase in performance when compared to the use of *ad hoc* estimates. The resulting hidden Markov model outperforms established methods like TileMap in the context of histone modification studies.

## 1 Background

High density oligonucleotide tiling arrays allow the investigation of transcriptional activity, protein-DNA interactions and chromatin structure across a whole genome. Tiling arrays have been used in a wide range of studies, including investigation of transcription factor activity [1] and of histone modifications in animals [2] and plants [3], as well as DNA methylation [4]. Analyses of these data are usually based either on a sliding window [1,5], or on hidden Markov models (HMMs) [6-8]. Other approaches have been suggested, e.g., by Huber *et al*. [9] and Reiss *et al*. [10], but are less common.

Parameter estimates for sliding window approaches as well as hidden Markov models are typically *ad hoc*. Although there are some notable exceptions in gene expression studies [8,11], no established procedures exist to obtain good parameter estimates from tiling array data, especially in the context of chromatin immunoprecipitation (ChIP-chip) experiments. Attempts have been made to obtain parameter estimates by integrating genome annotations into the analysis [12]. While this may provide good results when investigating transcriptional activity in well studied organisms, it is limited by the quality of available annotations. For ChIP-chip studies the required

annotation data is unavailable. A method for the localisation of transcription factors from ChIP-chip experiments by Keleş [13] does obtain the required parameter estimates from the data and allows for variations in length of enriched regions.

Methods designed for the analysis of ChIP-chip data focus almost exclusively on the study of transcription factors [[6,7,10], and [13]]. While this is an important class of experiments, ChIP-chip studies are not limited to transcription factors, and the analysis of other ChIP-chip experiments may require new methods. One other area of active research that utilises ChIP-chip experiments is the study of histone modifications and chromatin structure [3]. Although both types of experiment employ the same technology, there are several important differences between them. Most importantly, the 147 bp of DNA bound by a histone complex are considerably longer than the typical transcription factor binding site, and the histone modifications of interest are expected to affect several neighbouring histones. Consequently the ChIP fragments derived from a transcription factor binding site all originate from a small region containing the given binding site while regions affected by histone modifications can be much longer than the ChIP fragments used. As a result of this, the data from histone modification experiments usually contain long regions of interest encompassing several non-overlapping ChIP fragments, rather than the short and relatively isolated peaks produced by transcription factor studies.

Here we consider the analysis of data from a histone modification study in *Arabidopsis* [3]. These data consist of four ChIP samples for histone H3 with lysine 27 trimethylation (H3K27me3) and four histone H3 ChIP samples that act as a control. The aim of this analysis is to identify and characterise regions throughout the genome that exhibit enrichment for H3K27me3. It is desirable to use a method which is specifically designed for the analysis of histone modifications or flexible enough to accomodate the varying length of enriched regions. Furthermore, the method should obtain all parameter estimates from the data without the use of genome annotations and be robust towards outliers. Amongst the methods discussed above TileMap [7] comes closest to these requirements. Although it was developed with transcription factor analysis in mind it is general enough that it should provide useful results for other ChIP-chip experiments. This is emphasised by its application to histone modification [3] and DNA methylation [4] data as well as transtription factor analysis [14,15]. TileMap obtains some, but not all, of the required parameter estimates from the data. To provide a method which meets the requirements oulined above we develop a two state HMM with $t$ emission distributions. All parameter estimates for the model are obtained by maximum likelihood estimation using the Baum-Welch algorithm [16] and Viterbi training [17]. These methods have the advantage that no prior knowledge about parameter values is required and one need not rely on frequently unavailable genome annotations. To assess the performance of our model, we apply it to simulated and real data. Results are compared to those produced by TileMap. The remainder of this article is structured as follows. In Section 2 the hidden Markov model is developed and MLEs for all parameters are derived in Section 4. The performance of the resulting model is assessed in terms of sensitivity and specificity on simulated data in Sections 2.3.3–2.3.6. In Section 2.3.7 the model is used to analyse a public ChIP-chip data set [3] and results are compared to the original analysis of these data.

## 2 Results and discussion

Tiling array data consists of a series of measurements taken along the genome. Typically, microarray probes are designed to interrogate the genome at regular intervals. Design constraints such as probe affinity and uniqueness cause differences in probe density along the genome and can lead to large gaps between probes. Here we assume that the probe density is homogeneous except for a number of large gaps where the distance between adjacent probes is larger than max_gap. In the following analyses we use max_gap = 200 bp. This is identical to the value used by Zhang *et al*. [3], allowing for a direct comparison of results. Consider a ChIP-chip tiling array experiment with two conditions, a ChIP sample $\mathbf{X}_1$ targeting the protein of interest and a control sample $\mathbf{X}_2$. Each sample $\mathbf{X}_l$ has $m_l$ replicates ($l$ = 1, 2) providing measurements for $K$ genomic locations. The measurements for each probe are summarised by the "shrinkage $t$" statistic [18]:

$$\gamma_k = \frac{\bar{x}_{1k} - \bar{x}_{2k}}{\sqrt{\frac{v^*_{1k}}{m_1} + \frac{v^*_{2k}}{m_2}}}, \tag{1}$$

where $v^*_{lk}$ is a James-Stein shrinkage estimate of the probe variance obtained by calculating

$$v^*_{lk} = \hat{\lambda}^*_l s^2_{l\text{median}} + (1 - \hat{\lambda}^*_l) s^2_{lk}, \tag{2}$$

and $s^2_{lk}$ are the usual unbiased empirical variances and $\hat{\lambda}^*_l$ is the estimated optimal pooling parameter

$$\hat{\lambda}^*_l = \min\left(1, \frac{\sum_{k=1}^{K} \text{Var}(s^2_{lk})}{\sum_{k=1}^{K} (s^2_{lk} - s^2_{l\text{median}})^2}\right). \tag{3}$$

Other moderated $t$ statistics have been suggested and could be used instead, most notably the empirical Bayes $t$ statistic used by Ji and Wong [7] and the moderated $t$ of Smyth [19]. All of these approaches are designed to increase performance compared to the ordinary $t$ statistic by incorporating information from all probes on the microarray into individual probe statistics. Here we choose the "shrinkage $t$" because it does not require any knowledge about the underlying distribution of probe values while providing similar performance compared to more complex models [18].

### *2.1 Hidden Markov Model*

To detect enriched regions we use a two state discrete time hidden Markov model with continuous emission distributions and homogeneous transition probabilities (Figure 1), i.e., the transition probabilities depend only on the current state of the model. The use of homogeneous transition probabilities assumes equally-spaced probes within each observation sequence as well as a geometric distribution of the length of enriched regions. As discussed above there will be some variation in probe distances. Using a relatively small value for max_gap ensures that the assumption of homogeneity holds at least approximately. The two states of the model correspond to enrichment or no enrichment in the ChIP sample. The model is characterised by the set of states $\mathbf{S} = \{S_1, S_2\}$, the initial state distribution $p$, the matrix of transition probabilities $A$ and the state specific emission density functions $f_i$, $i = 1, 2$. The emission distribution of state $S_i$ is modelled as a $t$ distribution with location parameter $\mu_i$, scale parameter $\sigma_i$, and $\nu_i$ degrees of freedom.

The use of $t$ distributions has the advantage that their sensitivity to outliers can be adjusted via the degrees of freedom parameter, making them more robust and versatile than normal distributions. This is particularly useful when $\nu$ is estimated from the data [20]. It should be noted that the $\gamma_k$ modelled here are from a $t$-like distribution (Equation (1)). While this in itself might suggest the use of $t$ distributions for the $f_i$s, they are primarily chosen for their robustness. In the following we will refer to this model by its parameter vector $\theta = (\theta_1, \theta_2)$, where $\theta_1$ is the ordered pair $(p, A)$ and $\theta_2$ the ordered triple $(\mu, \sigma, \nu)$.

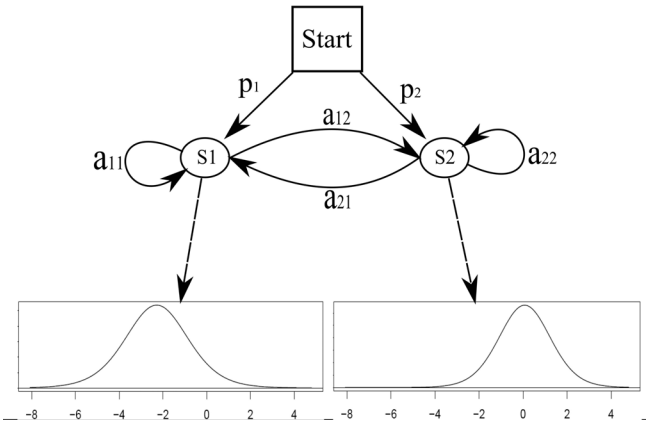


**Figure 1**
**Hidden Markov model for the analysis of ChIP-chip tiling array data.**

Given a hidden Markov model $\theta$ and an observation sequence $Y$, it is possible to compute the sequence of states $Q = q_1 q_2 ... q_K$ that is most likely to produce $Y$. There are several approaches to obtaining $Q$ [21]. Usually $Q$ is computed either by maximising the posterior probabilities $P(q_k = S_i|Y; \theta)$, $k = 1, ..., K$ or by calculating the sequence that maximises $P(Q|Y; \theta)$. The latter provides the single most likely sequence of states and can be computed efficiently by the Viterbi algorithm [22]. For the particular model used here both approaches are equivalent.

### *2.2 Parameter Estimation*

In this section we will discuss two different approaches to estimate $\theta$ for the model described in Section 2.1. The methods under consideration are the EM algorithm, which is usually known as the Baum-Welch algorithm in the context of HMMs, and Viterbi training. While the Baum-Welch algorithm is guaranteed to converge to a local maximum of the likelihood function, it is computationally intensive. Viterbi training provides a faster alternative but may not converge to a local maximum.

#### *2.2.1 Initial Estimates*

Both optimisation algorithms discussed here require initial parameter estimates. These are obtained from the data by first partitioning the vector of observations $Y$ into two clusters using $k$-means clustering [23]. From these clusters the location and scale parameters of the corresponding states are obtained as the mean and variance of the observations in the cluster. In the following, $\nu_1 = \nu_2 = 6$ is used as initial estimate for the degrees of freedom parameters.

#### *2.2.2 Baum-Welch Algorithm*

The Baum-Welch algorithm [16] is a well established iterative method for estimating parameters of HMMs. It represents the EM algorithm [24] for the specific case of HMMs. This algorithm can be used to optimise the transition parameters $\theta_1$ as well as the emission parameters $\theta_2$. Each iteration of the algorithm consists of two phases. During the first phase, the current parameter estimates are used to determine for each probe statistic in the observation sequence how likely it is to be produced by the different states of the model. In the second phase, parameters for the emission distributions of each state are estimated using contributions from all observations, according to the probability that they were produced by the respective

state of the model. The state transition parameters are updated in a similar fashion, accounting for the probability of transitions between states based on the observation sequence and the current model. After each iteration this procedure results in a model which explains the observed data better than the previous one, approaching a locally optimal solution. Using this method parameter estimates are updated until convergence is achieved. The details of the resulting algorithm are outlined in Section 4.1.

This method of parameter estimation is computationally expensive and time-consuming for a typical tiling array data set. The computing time can be reduced by fixing the degrees of freedom for the emission distributions in advance, thus avoiding the root-finding required for the estimation of these parameters. While this does not provide the same flexibility as estimating the required degree of robustness from the data it reduces the complexity of the optimisation problem. It is noted by Liu and Rubin [25] that attempts to estimate the degrees of freedom are more likely to produce results which are of little practical interest. The impact on classification performance of this choice is investigated in Section 2.3.

The formulation of the Baum-Welch algorithm used in this article is based on the description given by Rabiner [21] and on the EM algorithm derived by Peel and McLachlan [26] for fitting mixtures of *t* distributions.

#### *2.2.3 Viterbi Training*

While the Baum-Welch algorithm described in Section 2.2.2 is expected to provide good parameter estimates, it is computationally expensive. A faster model-fitting procedure can be devised by replacing the first phase of the Baum-Welch algorithm with a maximisation step. This method was introduced in [17] as segmental *k*-means and is now commonly referred to as Viterbi training. Unlike the Baum-Welch algorithm which allows each probe statistic to contribute to the parameter estimates for all states, Viterbi training assigns each observation to the state that is most likely to produce the given probe statistic. Thus each observation contributes to exactly one state of the model. While each iteration of this method is faster than one iteration of the Baum-Welch algorithm some iterations may decrease the likelihood of the model, thus failing to advance it towards a useful solution. See Section 4.2 for further details on the implementation of Viterbi training used here.

### *2.3 Testing*

#### *2.3.1 Simulated Data*

To assess the ability to distinguish between enriched and non-enriched probes of the models obtained by the different parameter estimation methods discussed in Section 2.2, we simulate data with known enriched regions. To ensure that the simulation study is providing meaningful results, it is based as closely on real data as possible. To this end, two independent analyses of the H3K27me3 data published by Zhang *et al*. [3] are carried out, one using TileMap [7], the other based on our model. The result of each analysis is used to generate a new dataset with known enriched regions. See Section 4 for further details. In the following these data are referred to as datasets I and II respectively. Since the simulation procedure is likely to bias results towards the model that was used in the process, we concentrate on the analysis of dataset I, with some results for dataset II presented for comparison. The use of data based on both models allows us to consider their performance under advantageous and disadvantageous conditions.

#### *2.3.2 Performance Measure*

The performance of different models on these data is determined in terms of false positive and false negative rates at probe level. While the relative importance of false positives and false negatives depends on the experiment under consideration, they are often equally problematic in the context of ChIP-chip experiments, especially when considering experiments which investigate differences between different cell lines or developmental stages, where all incorrect classifications are of equal concern. In this context, we define false positives as probes that are classified as non-enriched by the analysis of the real data but are called enriched in the subsequent analysis of simulated data, and vice versa for false negatives.

The output of each model is the estimated posterior probability of enrichment for each probe. In practice, probe calls ("enriched" or "non-enriched") are generated from this posterior probability based on a 0.5 cut-off. For any given model, classification performance will change with the chosen threshold. Thus we assess model performance across a range of cut-offs, reporting the relative number of false positives and false negatives as well as the error rate. The latter is also used to determine the cut-off that minimises incorrect classification results, and model performance is judged on the numbers of incorrect classifications at this optimal cut-off and at the usual 0.5 cut-off, and on the distance between the optimal cut-off and 0.5. The trade-off between sensitivity and specificity provided by the different models is characterised with ROC curves and the associated AUC values.

Another measure of interest is the ability to characterise the length distribution of enriched regions correctly. When studying chromatin structure the extent of structural changes is of interest; this is the case for the data studied in Section 2.3.7. This property of the different models is investigated in Section 2.3.6.

### *2.3.3 Estimating Degrees of Freedom*

We now consider the performance of both the Baum-Welch procedure and Viterbi training when all model parameters, including the degrees of freedom $\nu$, are estimated from the data. Both parameter estimation methods are used to fit an HMM to datasets I and II, and the performance of resulting models is assessed in terms of the achieved error rate (Figure 2), ROC curves (Figure 3) and their associated AUC (Table 1) for both datasets. To assess how well these methods perform in comparison to an established algorithm, we also fit a TileMap model to the two simulated datasets. The three models are compared to each other, as well as an *ad hoc* model which simply uses, without optimisation, the initial parameter estimates used by the two parameter optimisation methods. When comparing the performance of these models on both simulated datasets, it is important to consider that the simulation procedure introduces a bias towards the underlying model.

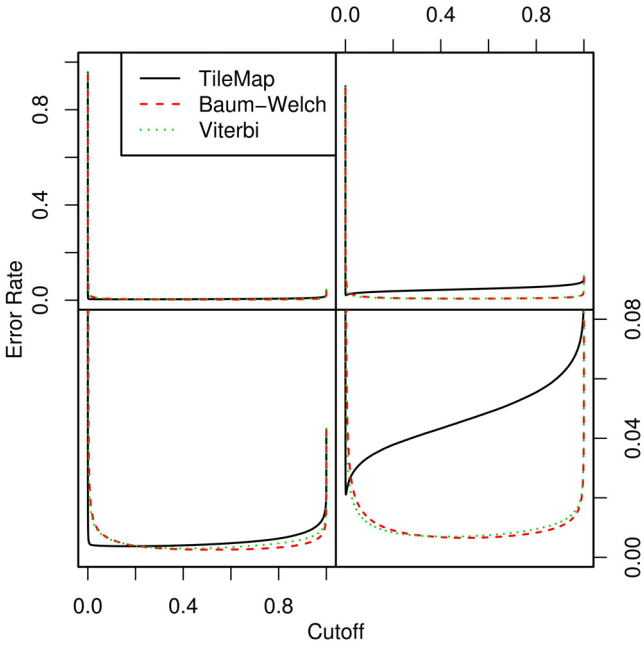


**Figure 2**
**Error rate for different models on datasets I and II.** Error rate resulting from the different models on dataset I (left) and II (right). When the total number of incorrect probe calls is considered, both parameter estimation procedures outperform TileMap on dataset I for cut-offs larger than 0.2. Both Baum-Welch and Viterbi training provide models with an optimal cut-off close to 0.5, while TileMap significantly underestimates the posterior probability resulting in an optimal cut-off of 0.19. The models with optimised parameters show similar performance on both datasets. On dataset II TileMap's performance is reduced in comparison to the results on dataset I. The main differences between the models considered here occur at error rates of 0–0.08. The relevant area of the figures in the top row is magnified in the plots below.

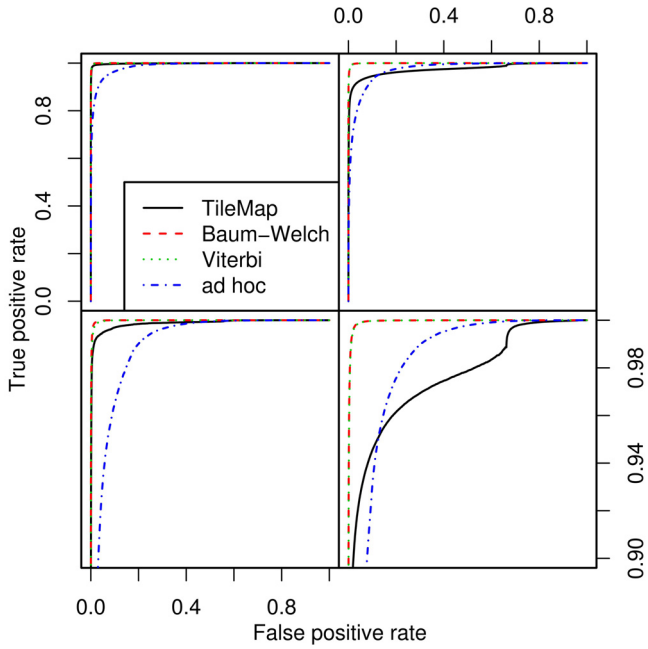


**Figure 3**
**ROC curves for different models on datasets I and II.** TileMap and the models with Baum-Welch and Viterbi training parameter estimates show similar performance on dataset I (left) with a small advantage for the models with optimised parameters. Comparison with a model using *ad hoc* parameter estimates highlights the performance increase achieved by optimising model parameters. On dataset II (right) TileMap performs similarly to the model with *ad hoc* parameter estimates. Figures on the bottom provide a close-up view of the plots above.

Estimating all parameters from the data with either the Baum-Welch algorithm or Viterbi training leads to models with high sensitivity, producing fewer false negatives than TileMap for any given cut-off [see Additional file 1]. At the same time they lead to an increased number of false positives [see Additional file 2] compared to TileMap, indicating a slight reduction of specificity. When considering the error rate it becomes apparent that both Baum-Welch and Viterbi training provide a favourable trade-off between sensitivity and specificity. These models reduce the number of incorrect classifications compared to TileMap both at the usual 0.5 cut-off and at the optimal cut-off. Moreover, while the Baum-Welch algorithm and Viterbi training both lead to models with an optimal cut-off close to 0.5 (0.51 and 0.42 respectively), TileMap provides an optimal cut-off of 0.19, indicating that it underestimates the posterior probability of enrichment. This becomes even more apparent when considering the result for dataset II where the optimal cut-off for TileMap is at 0.002

**Table 1: AUC for different models on both simulated datasets.**

| | TileMap | Baum-Welch | Viterbi-Training | Viterbi-EM | ad hoc |
|---|---|---|---|---|---|
| dataset I | 0.9986 | 0.9998 | 0.9997 | 0.9998 | 0.9869 |
| dataset II | 0.9749 | 0.9995 | 0.9994 | 0.9995 | 0.9728 |

All models with optimised parameters outperform TileMap on both simulated datasets. While TileMap performs well on dataset I it is only slightly better than the model with *ad hoc* parameter estimates.

compared to 0.5 for Baum-Welch and 0.41 for Viterbi training. This result suggests that TileMap is more tuned towards avoiding false positives than false negatives. From the above results we estimate that the weight given to false positives by TileMap is approximately 3.2 and 26 times larger than the weight for false negatives on datasets I and II respectively. The ROC curves (Figure 3) provide further evidence that the models with MLEs outperform TileMap. Although all three models perform well on dataset I, both parameter optimisation methods lead to better results than TileMap. The benefits of optimising parameter estimates are further highlighted by the performance of the model with ad hoc estimates that is used as starting point for the optimisation procedures. On both datasets, optimised parameters provide a notable increase in performance, with TileMap performing only slightly better than the ad hoc model on dataset II.

#### *2.3.4 Fixed Degrees of Freedom*

Estimating $\nu$, the degrees of freedom, for $t$ distributions from the data is time-consuming and may not be very accurate, especially for relatively large values of $\nu$. In this section we investigate the effect of fixing $\nu$ a priori for both states of the model. Only the case $\nu_1 = \nu_2$ is considered here. The remaining parameters are estimated from the training data using the Baum-Welch algorithm and Viterbi training with $\nu = 3, 4, ..., 50$. For each value of $\nu$, we report the error rate (Figure 4) as well as the AUC (Figure 5) on the simulated data.

For the best combination of $\nu$ and cut-off, both parameter estimation methods result in models with a classification performance comparable to the case of variable degrees of freedom (Figure 2). While the Baum-Welch algorithm tends to produce models with an optimal cut-off close to 0.5, Viterbi training only achieves this for large values of $\nu$. Notably, the best classification performance of the Viterbi trained model is achieved with 14 degrees of freedom and a 0.37 cut-off compared to 7 degrees of freedom and a 0.49 cut-off from Baum-Welch. This results in a decreased performance of the Viterbi model relative to the Baum-Welch model at the 0.5 cut-off.

#### *2.3.5 Convergence*

To reduce the time required for parameter estimation it is useful to limit the number of iterations. While each iteration of the Baum-Welch algorithm is guaranteed to improve the likelihood of the model, small changes to the parameter values do not necessarily lead to significant changes in the classification result. Furthermore, Viterbi training is not guaranteed to converge to a local maximum of the likelihood function and a likelihood based convergence criterion may not be appropriate for this method. Here we investigate the convergence of both algorithms based on the error rate and AUC to gauge the number of iterations required to achieve good classification results. Parameter estimation is performed with 60 iterations for both algorithms. Current estimates are used to classify the test data at every 5$^{th}$ iteration and AUC (Figure 5) and error rate (Figure 6) are determined.

The most striking difference in the convergence behaviour of the two methods is that Viterbi training appears to obtain good parameter estimates within a small number of iterations. Further iterations of the algorithm do not improve results substantially, whereas the Baum-Welch procedure provides parameter estimates that are better

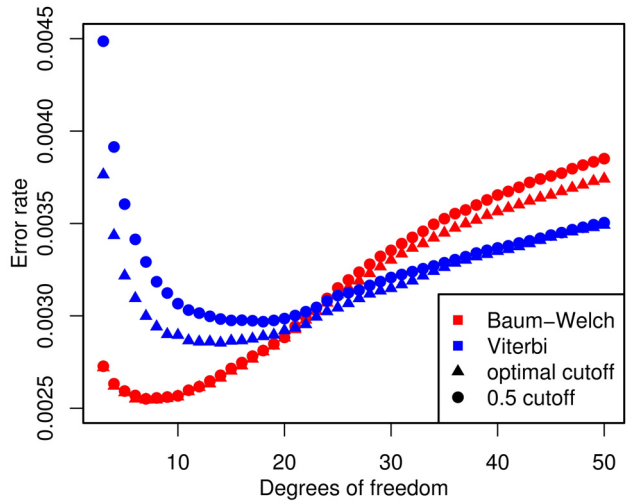


**Figure 4**
**Model performance for different choices of $\nu$.** The Baum-Welch model (red) performs better for relatively small values of $\nu$ while Viterbi training (blue) favours larger $\nu$. For the optimal choice of $\nu$ the Baum-Welch parameter estimates lead to an optimal cut-off close to 0.5.

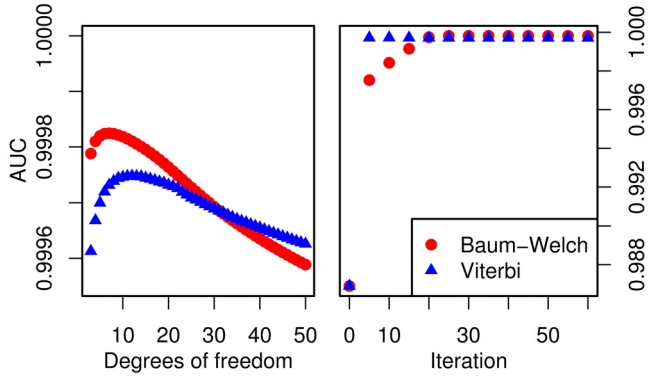


**Figure 5**
**AUC for different choices of $\nu$ and increasing numberof iterations**. Change in AUC for different choices of $\nu$ (left). The Baum-Welch model performs better for relatively small values of $\nu$ while Viterbi training favours larger $\nu$. Improvements in AUC with increasing number of iterations (right). The performance of the Viterbi trained model improves substantially during the first five iterations. Further iterations only produce small changes in the AUC. The Baum-Welch method requires more iterations to obtain the same AUC as as the Viterbi model. After 20 iterations the Baum-Welch model starts to outperform the Viterbi model.

than the ones obtained by Viterbi training, both in terms of likelihood and classification performance, but takes substantially longer to obtain these estimates. The Baum-Welch algorithm not only requires more iterations than Viterbi training, but the time required for each iteration is also longer.

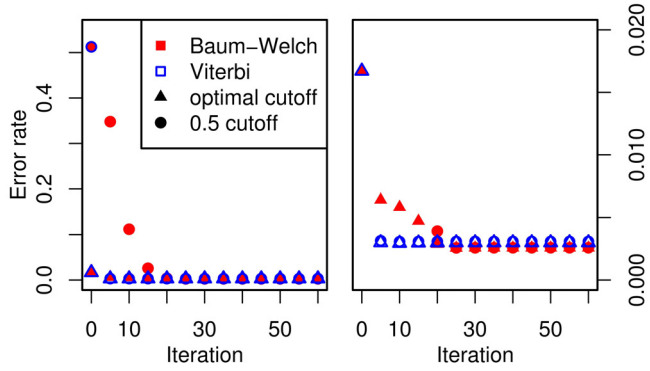


**Figure 6**
**Error rate at optimal and 0.5 cutoff for increasing number of iterations**. Parameter estimates obtained by the Baum-Welch algorithm (filled symbols) and Viterbi training (open symbols) improve model performance with increasing nuber of iterations. Viterbi training quickly approaches its optimal solution and initially outperforms Baum-Welch. The final model produced by the Baum-Welch algorithm provides a lower error rate than Viterbi training.

*2.3.6 Length Distribution of Enriched Regions*

When studying histone modifications one possible characteristic of interest is the length of enriched regions. To assess how accurately the different methods reflect the length distribution of enriched regions, we compare the length of regions predicted by TileMap and by the model (using Baum-Welch parameter estimates) to the length distribution of enriched regions in the simulated data (the "true length distribution"). Note that this length distribution may vary from the one found in real data. Nevertheless this comparison highlights some of the differences between the two models. Quantile-quantile plots of the respective length distributions show that TileMap systematically underestimates the length of enriched regions (Figure 7 (bottom left) and Figure 8 (bottom left)). While this effect is relatively small on dataset I there is some indication that it increases with region length and long regions may not be characterised appropriately by TileMap (Figure 7 (top left)). This observation is further supported by the length distribution of enriched regions produced by TileMap on dataset II (Figure 8 (left)). Enriched regions in dataset II are generally longer than regions in dataset I. This difference is not captured by

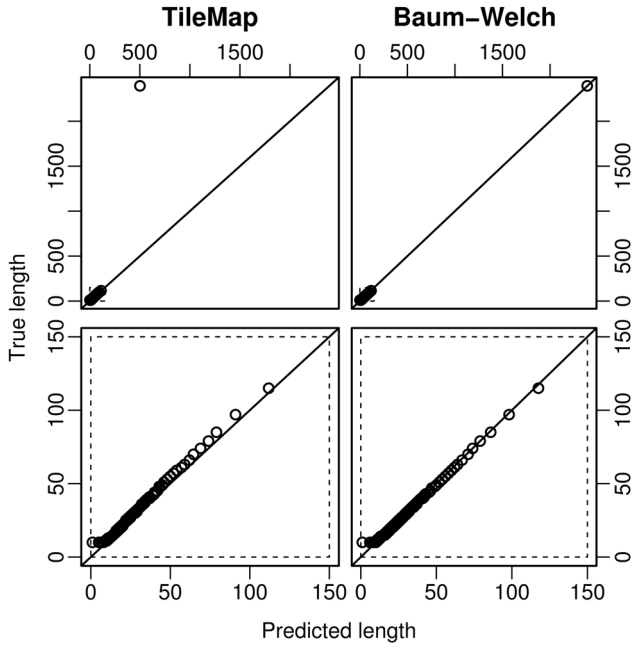


**Figure 7**
**Length distribution of enriched regions from dataset I**. Quantile-quantile plots comparing length distributions of enriched regions found with TileMap (left) and with the model based on maximum likelihood estimates (right) to the true length distribution of enriched regions in dataset I. Figures on the bottom provide a close-up view of the plots above. Each dot represents a percentile of the length distributions.

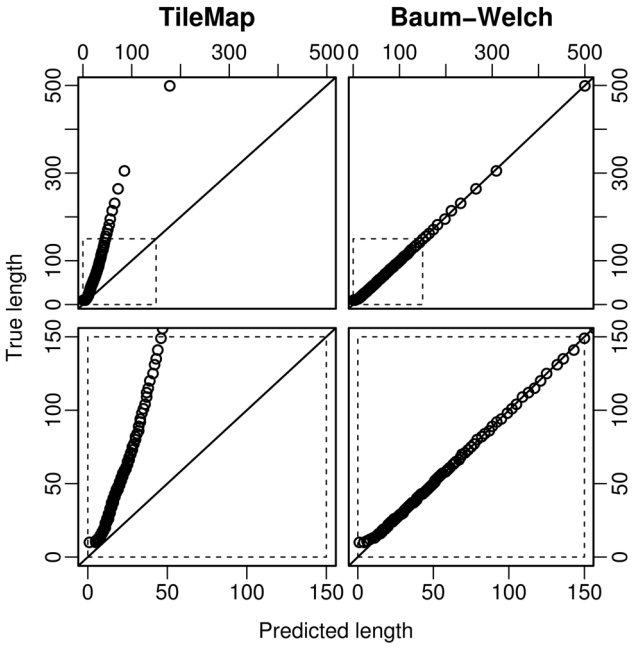


**Figure 8**
**Length distribution of enriched regions from dataset II**. Quantile-quantile plots comparing length distributions of enriched regions found with TileMap (left) and with the model based on maximum likelihood estimates (right) to the true length distribution of enriched regions in dataset II. Figures on the bottom provide a close-up view of the plots above. Each dot represents a percentile of the length distributions.

TileMap. Both TileMap and the Baum-Welch trained model produce several regions that are shorter than the shortest enriched region in the simulated data (Figure 7 (bottom)). There are two possible explanations for these short regions. They may be caused by underestimating the length of enriched regions, possibly splitting one enriched region into several predicted regions, or they may represent spurious enriched results produced by the model. In each case there is the possibility that the occurrence of extremely short regions is caused either by an intrinsic shortcoming of the model or by artifacts introduced during the simulation process. Since the simulation relies on TileMap to identify enriched and non-enriched probes it is inevitable that some probes will be misclassified. Subsequently these probes may be included in the simulated data, causing short disruptions of enriched and non-enriched regions. A sufficiently sensitive model could detect these unintended changes between enriched and non-enriched states.

To investigate further which of these is the case, we first examine the number of enriched probes contained in the short regions found by the Baum-Welch model and by TileMap respectively. The model with Baum-Welch parameter estimates found 126 regions with less than 10 probes. These regions contain a total of 866 probes of which 717 are in enriched regions. While this indicates that the majority of short regions is due to underestimating the length of enriched regions, several spurious probe calls remain. TileMap produced 249 regions with less than 10 probes, containing a total of 1781 probes, of which 1753 are in enriched regions. This is strong evidence that almost all of these short regions are caused by underestimating the length of enriched regions, and is consistent with the above observation that TileMap systematically underestimates the length of enriched regions.

To investigate whether the spurious short regions produced by the Baum-Welch model are due to an intrinsic shortcoming of the model or are artifacts introduced by the simulation procedure, we turn to real data. Here we focus on enriched regions containing only a single probe, which are most likely to be false positives. On dataset I the Baum-Welch model produced six of these extremely short regions. One of these probes is a true positive from an enriched region containing ten probes, i.e., the length of this region is underestimated by the Baum-Welch model. Of the remaining five probes three are identical, leaving three unique probes to be investigated further. For each of these three probes, we determine its position in the real data and its distance from enriched regions identified by TileMap and by our model (Section 2.3.7). Two of the probes are found to be located close to enriched regions identified by TileMap (142 and 391 bp) and all three probes are contained within enriched regions identified by our model [see Additional file 3]. This suggests that these probes may have been misclassified by TileMap during the original analysis, leading to an overestimation of the number of false positives produced by the Baum-Welch model on dataset I.

#### *2.3.7 Application to ChIP-Chip Data*

To investigate the performance of our model further, we apply it to the data of [3] and compare the result to the original analysis. Based on the results of the simulation study (Sections 2.3.3–2.3.6) we use the following procedure:

1. Quantile normalise and log transform data;

2. Calculate probe statistics (Equation (2));

3. Obtain initial estimates (Section 2.2.1);

4. Use 5 iterations of Viterbi training to improve initial estimates;

5. Use 15 iterations of Baum-Welch algorithm to obtain maximum likelihood estimates;

6. Apply resulting model to data to identify enriched regions.

This results in the detection of 5285 H3K27me3 regions covering 12.9 Mb of genomic sequence. Of these enriched regions, 3962 (~75%) are overlapping at least one annotated transcript. A total of 4982 or about 18.9% of all annotated genes are found to be enriched for H3K27me3. While most of the enriched regions cover a single gene, some regions are found to contain up to seven genes (Figure 9(b)). Enriched regions are predominantly longer than 1 kb with some extending over more than 20 kb (Figure 9(c)).

To assess whether there is a difference between regions of the genome that show H3K27me3 enrichment and the rest of the genome, we investigate the density of genes in the neighbourhood of genes that appear to be regulated by H3K27me3, and compare this to the gene density in other regions of the genome. For this purpose we obtain the gene density for the 50 kb upstream and downstream of each gene as (bp annotated as genes)/100 kb. The resulting gene densities for genes with and without enriched regions are summarised in Figure 9(a). There are visible differences between the two distributions which we test for significance with a two sided Kolmogorov-Smirnov test; this results in an approximate *p*-value of $2 \times 10^{-15}$. The significance of this result is further confirmed by a resampling experiment: the smallest *p*-value obtained from a series of 10000 resampled datasets is $1 \times 10^{-6}$.

## 3 Conclusion

With the use of MLEs for all model parameters, our model clearly improves classification performance on simulated data compared to *ad hoc* estimates, and outperforms TileMap. While our model produced some short regions that appear to be false positives, they are readily explained as a result of the simulation process. Comparison of results on simulated and real data suggests that TileMap produced a large number of false negatives in the original analysis used as the basis for the simulation. Inevitably, these false negatives were selected as part of non-enriched regions during the simulation process. The fact that the model with Baum-Welch parameter estimates was able to identify these isolated enriched probes despite the non-enriched contexts where they appeared emphasises the high sensitivity of the model.

TileMap's apparent tendency to penalise false positives more than false negatives clearly contributes to its relatively low performance in our comparisons which are based on the assumption that both types of error are equally problematic. While this is the case for the application considered here, one may argue that false positives are indeed of greater concern in some cases. When this is the case, TileMap's trade-off between sensitivity and specificity may lead to better results. However, it should be noted that the relative weights given to false positives and false negatives by TileMap can vary substantially between datasets. The parameter estimation procedure used for our model on the other hand provides consistent performance at the chosen cut-off.

The model-fitting procedure derived from the results of the simulation study (Sections 2.3.3–2.3.6) provides a fast and reliable approach to parameter estimation. This method retains all the favourable properties of the Baum-Welch algorithm while utilising the reduced computing time provided by Viterbi training. The use of MLEs ensures that model parameters are appropriate for the data. Results from the simulation study show that estimating

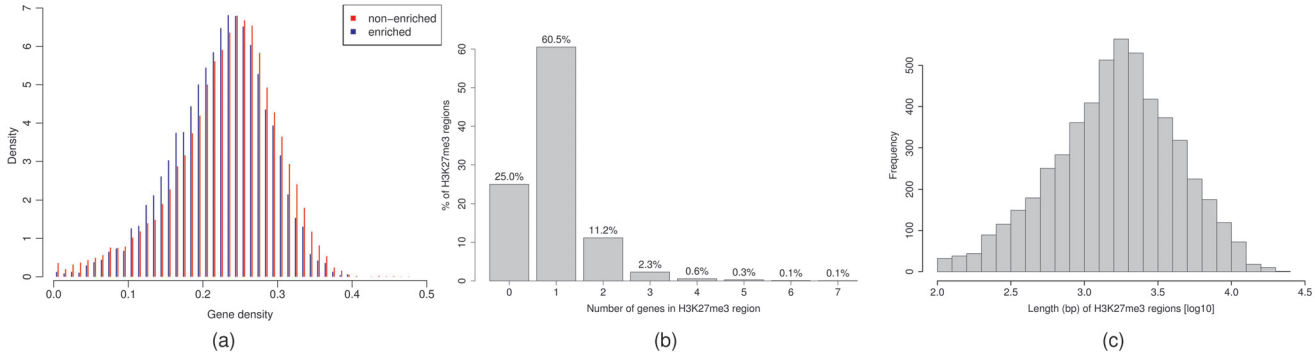


**Figure 9**

**Analysis of ChIP-chip data**. (a) Gene density in areas surrounding genes that contain H3K27me3 enriched regions and genes that do not contain enriched regions. (b) Number of genes found in H3K27me3 regions. While most enriched regions cover a single gene, there is a substantial number of H3K27me3 regions that cover several genes and enriched regions are found to contain up to seven genes. (c) Length distribution of H3K27me3 regions.

model parameters from the data improves the model's ability to recognise enriched regions of varying length and generally improves classification performance.

### *3.1 Future Work*

The analysis of the H3K27me3 data (Section 2.3.7) largely confirms the analysis of [3] although there are some notable differences. Most importantly, the H3K27me3 regions detected by our analysis are longer than the ones determined by TileMap (Figure 10). While Zhang *et al.* [3] found few regions longer than 1 kb, our analysis indicates that over 70% of enriched regions have a length of at least 1 kb, with the longest region spanning over 20 kb. Accordingly we find more regions that extend over several genes (Figure 9(b)). This may have implications for conclusions about the spreading of H3K27me3 regions in *Arabidopsis*.

At this stage, the biological significance of the observed difference in gene density in the neighbourhood of enriched and non-enriched genes is unclear. However, it indicates that the two groups of genes differ in a significant way. This suggests that the partition into enriched and non-enriched genes produced by our analysis is indeed meaningful.

The hidden Markov model presented in this article uses homogeneous transition probabilities, assuming that all probes are spaced out equally along the genome. To satisfy this assumption at least approximately, we use a fixed cut-off of 200 bp to partition the sequence of probe statistics such that there are no large gaps between probes. This arbitrary cut-off could be avoided by using a continuous time hidden Markov model.

## 4 Methods

### *4.1 Baum-Welch Algorithm*

The Baum-Welch algorithm [16] used to estimate parameters for our model is outlined in Section 2.2.2; further details are given below. Computing the likelihood of the long observation sequences produced by tiling arrays involves products of many small contributions. This typically results in likelihoods below machine precision. To avoid this effect computations are carried out in log-space, using the identity

$$\ln(x + y) = \ln(x) + \ln\left(1 + e^{\ln(y)-\ln(x)}\right). \quad (4)$$

In the following we use $^{\ln}\sum$ to denote summations which should be computed via Equation (4). The sequence of probe statistics $Y$ is split into $D$ observation sequences $Y^{(d)}$ such that the distance between probes within each observation sequence is at most max_gap and the distance between the end points of different observation sequences is greater than max_gap.

The emission distribution of state $S_i$ is given as

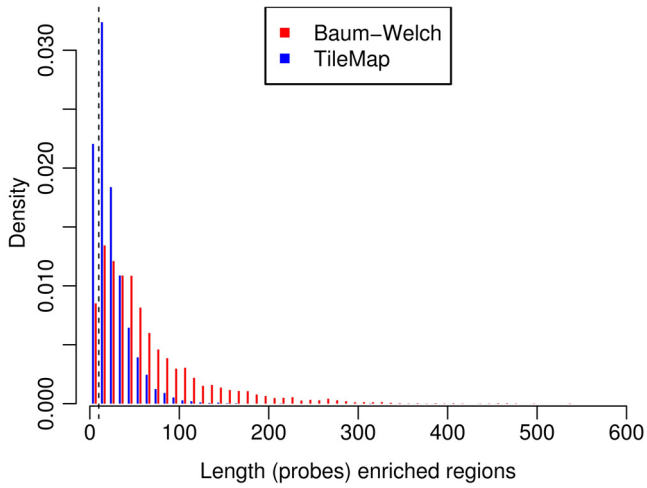


**Figure 10**
**Length distribution of enriched regions from real data**. Length distribution of enriched regions as determined by TileMap (blue) and Baum-Welch (red). Region length is determined in terms of probes per region. Both distributions were truncated at 10 for the simulation, ensuring that all regions in the simulated data contain at least ten probes.

$$f_i(\gamma_k;\mu_i,\sigma_i,\nu_i) = \frac{\Gamma\left(\frac{\nu_i+1}{2}\right)\Gamma\left(\frac{\nu_i}{2}\right)^{-1}}{\sigma_i\sqrt{\pi\nu_i}\left(1+\frac{(\gamma_k-\mu_i)^2}{\sigma_i\nu_i}\right)^{\nu_i+1}}. \quad (5)$$

For a given parameter set $\theta$ we can obtain new parameter estimates for transition probabilities by calculating

$$\xi^d_{kij} = \ln[P(q^d_k = S_i, q^d_{k+1} = S_j \mid Y^{(d)};\theta)] \quad (6)$$

$$= \alpha^d_{ki} + \ln[a_{ij}] + \ln[f_i(\gamma^d_{k+1};\theta_2)] + \\ +\beta^d_{(k+1)j} - \ln[P(Y^{(d)};\theta)]. \quad (7)$$

Here $\alpha_k$ and $\beta_k$ are known as forward and backward variables. For observation sequence $d$, $d$ = 1, ..., $D$, they are defined as

$$\alpha^d_{1i} = \ln[p_i] + \ln[f_i(\gamma^d_1;\theta_2)], \quad (8)$$

$$\alpha^d_{(k+1)j} = \left(\ln\sum_{i=1}^{N}(\alpha^d_{ki} + \ln[a_{ij}])\right) \\ +\ln[f_j(\gamma^d_{k+1};\theta_2)], \quad (9)$$

where $1 \le i \le N$, $1 \le j \le N$, $1 \le k < K_d$ and

$$\beta^d_{K_d i} = 0, \tag{10}$$

$$\beta^d_{ki} = \ln \sum_{j=1}^{N} \left( \ln[a_{ij}] + \ln[f_j(\gamma^d_{k+1}; \theta_2)] + \beta^d_{(k+1)j} \right), \tag{11}$$

where $1 \le i \le N$, $1 \le i \le N$, $k = K_d - 1, \ldots, 1$. Note that $\ln [P(Y^{(d)}; \theta)]$ is given by $\ln \sum_{i=1}^{N} \alpha^d_{K_d}$ We then calculate

$$\gamma^d_{ki} = \ln[P(q^d_k = S_i \mid Y^{(d)}; \theta)] \tag{12}$$

$$= \alpha^d_{ki} + \beta^d_{ki} - \ln[P(Y^{(d)}; \theta)] \tag{13}$$

$$= \ln \sum_{j=1}^{N} \xi^d_{kij}. \tag{14}$$

Combining the estimates from all observation sequences we obtain new parameter estimates for the transition probabilities:

$$\ln[a_{ij}] = \ln \sum_{d=1}^{D} \ln \sum_{k=1}^{K_d - 1} \xi^d_{kij} - \ln \sum_{d=1}^{D} \ln \sum_{k=1}^{K_d - 1} \gamma^d_{ki}, \tag{15}$$

$$\ln[p_i] = \ln \sum_{d=1}^{D} \gamma^d_{1i} - \ln[D]. \tag{16}$$

Calculations for the re-estimation of $\theta_2$ may involve negative values and cannot be carried out in log-space.

To obtain the required parameter estimates we first define $\ln[\tau^d_{ki}] = \gamma^d_{ki}$ and then compute

$$u^d_{ki} = \frac{\nu_i + 1}{\nu_i + (\gamma^d_k - \mu_i)^2}, \tag{17}$$

$$\hat{\mu}_i = \frac{\sum_{d=1}^{D} \sum_{k=1}^{K_d} \tau^d_{ki} u^d_{ki} \gamma^d_k}{\sum_{d=1}^{D} \sum_{k=1}^{K_d} \tau^d_{ki} u^d_{ki}}, \tag{18}$$

$$\hat{\sigma}_i = \frac{\sum_{d=1}^{D} \sum_{k=1}^{K_d} \tau^d_{ki} u^d_{ki} \left( \gamma^d_k - \hat{\mu}_i \right)^2}{\sum_{d=1}^{D} \sum_{k=1}^{K_d} \tau^d_{ki}}. \tag{19}$$

There is no closed form estimate for $\nu_i$. To obtain $\hat{\nu}_i$ one has to find a solution to the equation

$$\left[ -\psi\left(\frac{\nu_i}{2}\right) + \ln\left(\frac{\nu_i}{2}\right) + 1 + \right.$$
$$+ \frac{1}{\sum_{k=1}^{K_d} \tau^d_{ki}} \sum_{k=1}^{K_d} \tau^d_{ki} \left( \ln\left(u^d_{ki}\right) - u^d_{ki} \right) + \tag{20}$$
$$\left. + \psi\left(\frac{\nu_i + 1}{2}\right) - \ln\left(\frac{\nu_i + 1}{2}\right) \right] = 0$$

where $\psi$ is the digamma function. Standard root-finding techniques are employed to find a solution to (20).

### *4.2 Viterbi Training*

Viterbi training provides a faster alternative to the Baum-Welch algorithm. See Section 2.2.3 for a high level description of the algorithm. Details of the parameter estimation procedure are given below. Instead of calculating the conditional expectation of the complete data log likelihood, this algorithm first computes the most likely state sequence $Q$ given the observation sequence $Y$ and the current model $\theta$. The sequence $Y$ is partitioned according to $Q$, assigning each observation to the state that it most likely originated from. New estimates for $\theta_1$ are then obtained by calculating

$$\hat{p}_i = \frac{1}{D} \left| \{ d = 1, \ldots, D : q^d_1 = S_i \} \right|, \tag{21}$$

$$\hat{a}_{ij} = \frac{\left| \{ d=1,\ldots,D : q^d_k = S_i \text{ and } q^d_{k+1} = S_j \} \right|}{\sum_{d=1}^{D} (K_d - 1)}. \tag{22}$$

Updates for $\mu$ and $\sigma$ are obtained as in Section 2.2.1. The degrees of freedom $\nu$ can be either fixed in advance or estimated from the data using Equation (20) by setting $\tau^d_{ki} = 1$ if $(q^d_{ki}, q^d_{k+1}) = (S_i, S_j)$ and $\tau^d_{ki} = 0$ otherwise.

### *4.3 Simulated Data*

In a first step following the original analysis by [3], TileMap [7] is used with the HMM option to define enriched and non-enriched probes. Note that, although this classification of probes is not perfect, it can be assumed that most probes are assigned to the correct group. The length distribution of enriched and non-enriched regions detected by TileMap is used to determine the length distributions for the simulated data after removing all regions that contain less than 10 probes (Figure 10). Data are generated by first determining the length of enriched and non-enriched regions from the empirical length distributions and then sampling data points from

the respective TileMap generated clusters. Following this procedure, 600 sequences with one to ten enriched regions in each sequence are generated. A second dataset is generated by applying the model described in Section 2. Note that, although this procedure relies on the classifications produced by the respective models, the resampling procedure will place individual probe values in a new context of surrounding probes, which may lead to different probe calls in the analysis of the simulated data. Prior to analysis all data are quantile normalised.

## 5 Availability

The parameter estimation methods used in this article are available as part of the R package tileHMM from the authors' webpage http://www.bioinformatics.csiro.au/TileHMM/ and from CRAN. The simulated data used in this study is available from the authors' web page.

## 6 Authors' contributions

PH conducted the research and wrote the manuscript. DB critically revised the manuscript. GS conceived the project. DB and GS provided supervision to PH. All authors have read and approved the final manuscript.

## Additional material

### Additional file 1

***False negative probe calls resulting from different models.*** *For any given cut-off TileMap produces more false negatives than the Baum-Welch and Viterbi trained models.*

Click here for file
[http://www.biomedcentral.com/content/supplementary/1471-2105-9-343-S1.png]

### Additional file 2

***False positive probe calls resulting from different models.*** *For any given cut-off TileMap produces fewer false positives than the Baum-Welch and Viterbi trained models.*

Click here for file
[http://www.biomedcentral.com/content/supplementary/1471-2105-9-343-S2.png]

### Additional file 3

***Origin of isolated enriched probes in dataset I.*** *The isolated enriched probes identified in dataset I by the Baum-Welch model originate from enriched regions identified by the Baum-Welch model in the real data. Two out of three probes are located close to enriched regions identified by TileMap.*

Click here for file
[http://www.biomedcentral.com/content/supplementary/1471-2105-9-343-S3.png]

## Acknowledgements

PH is supported by an MQRES scholarship from Macquarie University and a top-up scholarship from CSIRO. The authors would like to thank Michael Buckley for his helpful suggestions.

## References

1. Cawley S, Bekiranov S, Ng HH, Kapranov P, Sekinger EA, Kampa D, Piccolboni A, Sementchenko V, Cheng J, Williams AJ, Wheeler R, Wong B, Drenkow J, Yamanaka M, Patel S, Brubaker S, Tammana H, Helt G, Struhl K, Gingeras TR: **Unbiased Mapping of Transcription Factor Binding Sites along Human Chromosomes 21 and 22 Points to Widespread Regulation of Noncoding RNAs.** *Cell* 2004, **116:**499-509.
2. Bernstein BE, Kamal M, Lindblad-Toh K, Bekiranov S, Bailey DK, Huebert DJ, McMahon S, Karlsson EK, III EJK, Gingeras TR, Schreiber SL, Lander ES: **Genomic Maps and Comparative Analysis of Histone Modifications in Human and Mouse.** *Cell* 2005, **120:**169-181.
3. Zhang X, Clarenz O, Cokus S, Bernatavichute YV, Goodrich J, Jacobsen SE: **Whole-Genome Analysis of Histone H3 Lysine 27 Trimethylation in Arabidopsis.** *PLoS Biol* 2007, **5(5):**e129.
4. Zhang X, Yazaki J, Sundaresan A, Cokus S, Chan SWL, Chen H, Henderson IR, Shinn P, Pellegrini M, Jacobsen SE, Ecker JR: **Genome-wide High-Resolution Mapping and Functional Analysis of DNA Methylation in Arabidopsis.** *Cell* 2006, **126:**1189-1201.
5. Bertone P, Stolc V, Royce TE, Rozowsky JS, Urban AE, Zhu X, Rinn JL, Tongprasit W, Samanta M, Weissman S, Gerstein M, Snyder M: **Global Identification of Human Transcribed Sequences with Genome Tiling Arrays.** *Science* 2004, **306(5705):**2242-2246.
6. Li W, Meyer CA, Liu XS: **A hidden Markov model for analyzing ChIP-chip experiments on genome tiling arrays and its application to p53 binding sequences.** *Bioinformatics* 2005, **21(Suppl 1):**i274-i282.
7. Ji H, Wong WH: **TileMap: create chromosomal map of tiling array hybridisations.** *Bioinformatics* 2005, **21(18):**3629-3636.
8. Munch K, Gardner PP, Arctander P, Krogh A: **A hidden Markov model approach for determining expression from genomic tiling micro arrays.** *BMC Bioinformatics* 2006, **7:**239.
9. Huber W, Toedling J, Steinmetz LM: **Transcript mapping with high-density oligonucleotide tiling arrays.** *Bioinformatics* 2006, **22(16):**1963-1970.
10. Reiss DJ, Facciotti MT, Baliga NS: **Model-based deconvolution of genome-wide DNA binding.** *Bioinformatics* 2008, **24(3):**396-403.
11. Toyoda T, Shinozaki K: **Tiling array-driven elucidation of transcriptional structures based on maximum-likelihood and Markov models.** *The Plant Journal* 2005, **43:**611-621.
12. Du J, Rozowsky J, Korbel JO, Zhang ZD, Royce TE, Schultz MH, Snyder M, Gerstein M: **A supervised hidden Markov model framework for efficiently segmenting tiling array data in transcriptional an ChIP-chip experiments: systematically incorporating validated biological knowledge.** *Bioinformatics* 2006, **22(24):**3016-3024.
13. Keleş S: **Mixture Modeling for Genome-Wide Localization of Transcription Factors.** *Biometrics* 2007, **63:**10-21.
14. Ji H, Vokes SA, Wong WH: **A comparative analysis of genome-wide chromatin immunoprecipitation data for mammalian transcription factors.** *Nucl Acids Res* 2006, **34(21):**e146.
15. Sandmann T, Girardot C, Brehme M, Tongprasit W, Stolc V, Furlong EEM: **A core transcriptional network for early mesoderm development in Drosophila melanogaster.** *Genes & Development* 2007, **21(4):**436-449.
16. Baum LE, Petrie T, Soules G, Weiss N: **A Maximization Technique Occuring in the Statistical Analysis of Probabilistic Functions of Markov Chains.** *The Annals of Mathematical Statistics* 1970, **41:**164-171.
17. Juang BH, Rabiner LR: **A segmental k-means algorithm for estimating parameters of hidden Markov models.** *IEEE Transactions on Acoustics, Speech, and Signal Processing* 1990, **38(9):**1639-1641.
18. Opgen-Rhein R, Strimmer K: **Accurate Ranking of differentially expressed genes by a distribution-free shrinkage approach.** *Statistical applications in Genetics and Molecular Biology* 2007, **6:**Article 9.
19. Smyth GK: **Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments.** *Statistical Applications in Genetics and Molecular Biology* 2004, **3:**Article 3.
20. Lange KL, Little RJA, Taylor JMG: **Robust Statistical Modeling Using the t Distribution.** *Journal of the American Statistical Association* 1989, **84(409):**881-896.
21. Rabiner LR: **A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.** *Proceedings of the IEEE* 1989, **77(2):**257-286.

22. Viterbi AJ: **Error bounds for convolutional codes and an asymptotically optimal decoding algorithm.** *IEEE Transactions on Information Theory* 1967, **13:**260-269.
23. Hartigan JA, Wong MA: **A K-means clustering algorithm.** *Applied Statistics* 1979, **28:**100-108.
24. Dempster AP, Laird NM, Rubin DB: **Maximum Likelihood for Incomplete Data via the EM Algorithm.** *Journal of the Royal Statistical Society, Series B* 1977, **39:**.
25. Liu C, Rubin DB: **ML estimation of the t distribution using EM and its extensions, ECM and ECME.** *Statistica Sinica* 1995, **5:**19-39.
26. Peel D, McLachlan GJ: **Robust mixture modelling using the t distribution.** *Statistics and Computing* 2000, **10:**339-348.