BIOINFORMATICS ORIGINAL PAPER Vol. 26 no. 11 2010, pages 1423–1430
doi:10.1093/bioinformatics/btq162

Gene expression Advance Access publication April 15, 2010

A hidden Markov support vector machine framework incorporating profile geometry learning for identifying microbial RNA in tiling array data

Wen-Han Yu[1,2], Hedda Høvik[3] and Tsute Chen[1,*]

[1]Department of Molecular Genetics, The Forsyth Institute, Boston, MA 02115, [2]Bioinformatics Graduate Program, Boston University, Boston, MA 02118, USA and [3]Department of Oral Biology, Faculty of Dentistry, University of Oslo, Oslo, Norway

Associate Editor: Ivo Hofacker

ABSTRACT

Motivation: RNA expression signals detected by high-density genomic tiling microarrays contain comprehensive transcriptomic information of the target organism. Current methods for determining the RNA transcription units are still computation intense and lack the discriminative power. This article describes an efficient and accurate methodology to reveal complicated transcriptional architecture, including small regulatory RNAs, in microbial transcriptome profiles.

Results: Normalized microarray data were first subject to support vector regression to estimate the profile tendency by reducing noise interruption. A hybrid supervised machine learning algorithm, hidden Markov support vector machines, was then used to classify the underlying state of each probe to 'expression' or 'silence' with the assumption that the consecutive state sequence was a heterogeneous Markov chain. For model construction, we introduced a profile geometry learning method to construct the feature vectors, which considered both intensity profiles and changes of intensities over the probe spacing. Also, a robust strategy was used to dynamically evaluate and select the training set based only on prior computer gene annotation. The algorithm performed better than other methods in accuracy on simulated data, especially for small expressed regions with lower (<1) SNR (signal-to-noise ratio), hence more sensitive for detecting small RNAs.

Availability and implementation: Detail implementation steps of the algorithm and the complete result of the transcriptome analysis for a microbial genome *Porphyromonas gingivalis* W83 can be viewed at http://bioinformatics.forsyth.org/mtd

Contact: tchen@forsyth.org

Received on 15 January 2010; revised on 22 March 2010; accepted on 9 April 2010

1 INTRODUCTION

Current microarray manufacturing technology can synthesize millions of oligonucleotide probes *in situ* on a single microscopic glass slide. This great capacity of probes allows the design of genomic tiling microarrays containing probes covering both sense and antisense strands with a great frequency for most microbial genomes (Akama *et al.*, 2009; Selinger *et al.*, 2000; Tjaden *et al.*, 2002) as well as many eukaryotic genomes (Bertone *et al.*, 2004; David *et al.*, 2006; Kapranov *et al.*, 2002; Li *et al.*, 2007; Schadt *et al.*, 2004; Selinger *et al.*, 2000; Stolc *et al.*, 2004; Yamada *et al.*, 2003). These high-density genomic tiling arrays can be used to detect the expression for all RNA species including protein coding RNAs and non-coding RNAs (ncRNAs).

Although becoming popular, the data analysis for expression data obtained from the high-density tiling microarrays remains to be challenging. A fundamental task is how to precisely identify expression from noisy background. One common approach has been to segment the probe signals along the genomic coordinates. The assumption of this approach is that intensities within a transcript distribute as Gaussian noise. Thus, the breakpoints detected by the abrupt changes between two adjacent segments may represent the boundaries of the RNA transcripts. A segmentation algorithm (Bai and Perron, 2003; Huber *et al.*, 2006; Picard *et al.*, 2005) was developed to model the signal distribution that fitted Gaussian noise. Finding an optimal set of breakpoint locations that minimized the sum of squared residuals was accomplished by the dynamic programming algorithm with a fixed number of segments. However, it is not robust to estimate the total number of segments. Also, long computation time $O(n^2S)$ was required, where S is the number of segments and n is the genome size. For this, Huber *et al.* (2006) attempted to simplify the complexity of the algorithm with fixed maximum length l of segment and reduced the computation time to $O(nlS)$. The determination of expression status for each segment relied, however, on a fixed cutoff value (David *et al.*, 2006), which may overlook RNA with low expression level.

A supervised learning algorithm using Hidden Markov Models (HMMs) has been introduced to directly distinguish transcribed and non-transcribed regions (Du *et al.*, 2006; Li *et al.*, 2005; Munch *et al.*, 2006). This approach successfully incorporated validated biological knowledge into the model, instead of only considering the signal distribution that might be biased by noises or system errors. By given a training set, HMMs constructs a probabilistic model that connects the hidden states to the observables as well as to the adjacent states. Viterbi algorithm is then used to compute the most likely hidden state sequence. However, higher order HMMs are known to be a better model to describe the dependency between neighbors than typical first-order HMMs, although it has not been applied widely due to the complexity and computational demands. In addition, several limitations of the conventional HMMs have

*To whom correspondence should be addressed.

been noted, such as that it typically trains a conditional probabilistic model rather than a discriminative hyperplane, and it lacks the power of processing highly dimensional feature vectors which represent the observed input sequence (Rabiner, 1989).

A different classification algorithm, support vector machines (SVMs) (Cortes and Vapnik, 1995) providing a discriminative model, has been commonly used in various biological applications, such as the classification of the normal and cancer tissues from microarray expression data (Furey *et al.*, 2000). SVMs are capable of finding a linear discriminative hyperplane for classification in a high-dimensional feature space, projected from the input object space by either linear or non-linear mapping via the kernel functions. However, the classification of individual object only estimates from the corresponding feature vectors. This approach is inappropriate for interpreting the transcriptome data directly because conditional dependency of the neighbors along the sequence needs to be considered as well.

To take advantage of both learning algorithms, we implemented a novel discriminative algorithm 'hidden Markov support vector machines' (HM-SVMs). HM-SVMs combines HMMs and SVMs (Altun *et al.*, 2003; Joachims *et al.*, 2009; Zeller *et al.*, 2008) and retains the Markov chain dependency structure between the hidden states as well as the efficiency of dynamic programming by Viterbi algorithm. Additional important components inherited from SVMs are also retained. The discriminative hyperplane is learned by kernel functions and determined by the maximum-margin principle with soft margin violation adjustment. At the same time, HM-SVMs show the capability to handle high-dimensional feature vectors and overlapping features. As a result, the input data of the HM-SVMs for feature vector construction can be readily extended through incorporating multiple experimental validated data. Zeller *et al.* (2008) implemented a similar SVM discriminative technique to analyze *Arabidopsis thaliana* tiling data. They modeled the exon–intron expression mechanism specific to the eukaryotes, which may not be suitable to the operon expression mechanism of the prokaryotes. In addition, recent tiling array probe design methods generate probe sets with unequal probe spacing (Hovik and Chen, 2010), therefore a probabilistic model that considers both intensity profile and change of intensity across the probe location will be more adequate for describing the transcriptome profile detected with such probe design. In this study, we constructed a heterogeneous HMM model with profile geometry learning to include both intensity profile and positional changes. Together with normalization and dynamic training data screening, we present a comprehensive and robust methodology for predicting the occurrence of the transcription units across the genomic sequence on both strands from the tiling array expression data. We used this new method to study the architecture and dynamics of transcription activity of a model organism, *Porphyromonas gingivalis* W83, which is an important periodontal pathogen.

2 MATERIALS AND METHODS

2.1 Experimental data

Microarrays used in this study were fabricated by Roche NimbleGen, Inc. (Madison, WI, USA) and each contained 385 000 unique 50mer sequences covering both sense and antisense strands of the entire genome of *P.gingivalis* W83 at a frequency of ca. one probe per 12 bases in average. Probe sequences were designed by using a dynamic genomic tiling array probe design pipeline (Hovik and Chen, 2010). The probe set can be downloaded from http://bioinformatics.forsyth.org/mtd.

Total RNA and genomic DNA were extracted from *P.gingivalis* W83 grown on TSA sheep blood agar plates containing Hemin and Vitamin K (BAPHK) for 2 days in an anaerobic chamber at 37°C (Duncan *et al.*, 1993), and were labeled with Label IT μArray Cy3 Labeling Kits (Mirus Bio LLC, Madison, WI, USA). Microarray hybridizations were performed at 42°C for 16 h in the chamber with the Long Oligo hybridization buffer [80 mM Tris–HCl, pH 7.0, 8 mM ethylenediaminetetraacetic acid (EDTA), 25% formamide, 5× SSC (75mM Trisodium Citrate, 750mM Sodium Chloride), 0.1% sodium dodecyl sulfate, 0.7 mg/ml salmon sperm DNA]. Post-hybridization procedures including array washing and signal acquisition were done according to Nimblegen's manufacturer protocol. Three biological replicates for both RNA and DNA samples were used for data analysis.

2.2 Data normalization

Raw microarray intensities were adjusted with data from DNA reference arrays using the Bioconductor package 'tilingArray' under R statistical programming environment (Huber *et al.*, 2006). A variety of factors affect the range of hybridization signals including different thermodynamic properties imposed by probe sequences (Royce *et al.*, 2005), biases in labeling efficiency and the abundance of target sequences. The abundance of target molecule y'_{ij} can be modeled as,

$$y'_{ij} = \frac{y_{ij} - B_i}{A_i} \tag{1}$$

where y_{ij} is the observed intensity of i-th probe on the j-th array, B_i is the unspecific background fluorescence and A_i is the proportional factor specific to the abundance of y'_{ij}. The unknown parameters A_i and B_i were estimated directly or indirectly from the corresponding genomic DNA reference intensities. Non-specific background for B_i was estimated from 80% of the probes with lowest intensities in the intergenic regions of the genome. The probe intensities with repeated sequences in the genome were regressed to the level equivalent to that of a single copy of the sequence. Between-array normalization (Huber *et al.*, 2002) included in the tilingArray package was used for adjusting systematic signal variations among the arrays and base 2 logarithm of probe intensities were scaled.

2.3 Signal noise reduction by support vector regression normalization

We applied SVMs (Cortes and Vapnik, 1995) to intensity regression implemented by the 'kernlab' package in R (Karatzoglou *et al.*, 2004). The SVMs is a kernel-based machine learning algorithm. It maps the input data x into a high-dimensional feature space H defined by a kernel function and then searches for a hyperplane, i.e. a linear relation $f(x)$, among the data points in the feature space:

$$f(x) = < w, \Phi(x) > + b \tag{2}$$

where $\Phi(x)$ is projection $\Phi : x \to H$ by the corresponding feature vector $x \in \mathbb{R}^n$; w is the weight vector perpendicular to the hyperplane and weighting the corresponding dimension, and $b \in \mathbb{R}$. The SVMs were developed to solve pattern classification (support vector classification) and regression problem [support vector regression (SVR)]. The SVR has been extensively implemented in studying financial time series prediction (Huang *et al.*, 2006).

2.3.1 The model In this study, the model was learned by giving the input data, which in our case was the hybridization measurements $\{(x_1, y_1), (x_2, y_2) \ldots (x_i, y_i)\}$, where x_i is the probe genomic coordination and y_i the normalized intensity. To perform regression, SVR uses a different loss function compared to common SVMs called ε-insensitive loss function:

$$|y - f(x)| = \begin{cases} 0, & \text{if } |y - f(x)| < \varepsilon \\ |y - f(x)| < \varepsilon & \text{otherwise} \end{cases} \tag{3}$$

Only the data points larger than the threshold of $\pm\varepsilon$ from the predicted linear function $f(x)$ were subjected to the penalty. The complete optimization of SVR minimizes sum of the loss function and regularization item below:

$$\begin{aligned} &\text{Minimize} \quad t(w,\xi)=C\textstyle\sum_{i=1}^{m}(\xi_i+\xi_i^*)+\frac{1}{2}||w||^2 \\ &\text{Subject to} \quad (<w,\Phi(x_i)>+b)-y_i\le\varepsilon-\xi_i \\ &\qquad\qquad y_i-(<w,\Phi(x_i)>+b)\le\varepsilon-\xi_i^* \\ &\qquad\qquad \xi_i\ge 0,\ \xi_i^*\ge 0,\ i=1,2,...,m, \end{aligned} \tag{4}$$

where m is the number of total probes on the genome; C is the trade-off value, and ξ_i and ξ_i^* are the corresponding positive and negative errors at the i-th probes, respectively. The epsilon tube around the decision line of $f(x)$ was created by the loss function. Therefore, only the data points outside the tube area had impact on the final decision line and the degree of influence was determined by the distance to the decision line.

2.4 RNA transcripts identification by HM-SVMs

HM-SVMs was used to decode transcribed (expressed) and non-transcribed (silent) regions from the complicated probe intensity profiles. This hybrid algorithm allowed labeling the hidden state of sequential data based on the model learned from a training dataset integrating with prior knowledge.

2.4.1 The model To refine the problem, the transcriptome data D from hybridization measurements consisting of probes $x=(x_1, x_2, \ldots, x_m,)$ are associated with the unknown hidden state sequence $e=(e_1, e_2,\ldots,e_m)$ generated by an unknown model M. The feature vector $V\equiv \{x_i=(v_1, v_2,\ldots,v_n)\}i=1\ldots m\in\mathbb{R}^n$, where x_i is i-th probe on the genome and v is one of n features, was constructed from D to characterize every probe. In order to predict the unknown state e over the sequence (i.e. expressed or silent states), we constructed a learning model M' representing a true model M. The constraint data $\mathcal{X}\equiv\{(x_i, e_i)\}\in D$, in which the hidden state sequence $e_{\text{constraint}}$ has been characterized by prior knowledge, were selected as a training dataset for algorithm modeling. A w-parameterized discriminative function F: $\mathcal{X}\times \text{E}\in\mathbb{R}$ was generated by maximizing Fover the response variable $e_{\text{constraint}}\in E$ for a specific input $\mathcal{X}$:

$$f(x)=\arg\max F(\mathcal{X},e_{\text{constraint}};w) \tag{5}$$

The optimized function conjugated the pair of observation and hidden state sequences by a mapping Φ, which extracted the features from observation/hidden state sequence pairs (x, e):

$$F(\mathcal{X},e_{\text{constraint}};w)= <w,\Phi(x,e)> \tag{6}$$

Suggested by the concept of HMM, the hidden state of probe intensity along the genome was considered as a Markov chain. Two types of the features that jointed input–output mapping were derived from the emission and transition matrices. The emission matrix combined attributes of the observation vectors with a specific hidden state. The transition matrix described the neighbor hidden states which depended on each other along the sequence. Therefore, the function F was rewritten as following:

$$\begin{aligned} &F(\mathcal{X},e_{\text{constraint}};w)= \\ &\textstyle\sum_{i=1}^{T}\sum_{\sigma,\tau\in K}[[e^{i-1}=\sigma\wedge e^i=\tau]]+[[e^i=\tau]]\psi_y(x^i) \end{aligned} \tag{7}$$

where $[[e^i=\tau]]$ $\psi_y(x^i)$ represents combination of hidden states and observation in which $\psi_y(x^i)$ maps observation vector associated with the observation r from the i-th point in the sequence. In our case, $\psi_y(x^i)$ denotes the observed intensity $y\in\mathbb{R}$ at i-th probe in the sequence. And $[[e^i=\tau]]$ shows an indicator function for the hidden state τ located at the i-th probe. $[[e^{i-1}=\sigma\wedge e^i=\tau]]$ displays the dependencies of states $\sigma,\tau\in K$ at the $(i-1)$-th and i-th positions, where K denotes all possible states. In our case, K is either expressed or silent state. The equation mentioned above corresponds to first-order models, and higher order models can be generalized as well. $F(\mathcal{X},e_{\text{constraint}};w)$ accumulates all extracted features along the sequence of length T. The maximum separation margin defined by the kernel function from the training data points in the feature space, which minimized errors of misclassification, was used to construct the discriminative function F. As described before, to solve possible non-separable data points, ξ slack variables and error cost C controlling the trade off were introduced to create a soft margin to allow margin violations. More detail HM-SVMs optimization combining the loss function with regularization item was described in Altun *et al*. (2003). As a result, the hidden state sequence e' was estimated by the function F, giving a complete test data with associated feature vector.

2.4.2 Extraction of feature vector We introduced a profile geometry learning method to construct feature vectors. In order to depict the profile terrain shaped by probe intensity and position, the feature vectors were composed of two terms including elevation and change of slopes. Given a search window (the range of the flanking regions of the current probe j) $S=(x_i, x_{i+1},\ldots,x_s)$ which represents the probe j, its feature vector can be written as $V_j\equiv \{(f(\varphi)_{i,i+1}f(\varphi)_{i+1,i+2,}..f(\varphi)_{s-1,s}), (z_i, z_{i+1,}\ldots,z_s)\}$. z_i is the Z-score converted from the probe intensity under the assumption of the probe signals distributed by Gaussian. Therefore, Z-score normalizes the signals and presents the elevation of the terrain. $f(\varphi)_{i,i+1}$ shows the score of jumping from probe x_i to x_{i+1} as below:

$$f(\varphi)_{i,i+1}=\left(\frac{\Delta y_{i,i+1}}{\Delta x_{i,i+1}}\right)\left(\frac{O(\Delta y)_{\Delta x}}{E(\Delta y)_{\Delta x}}\right) \tag{8}$$

where y and x showed the probe intensity and position. The first factor is the slope between probe x_i, x_{i+1}; the second was a weight which compared $O(\Delta y)_{\Delta x}$ (the observed y difference at the distance of x_i and x_{i+1}) to $E(\Delta y)_{\Delta x}$ (the expected y difference at the same distance). $E(\Delta y)_{\Delta x}$ can be estimated by averaging Δy calculated from all probe distances. Note that the probes were not equally distributed on genomic sequence. $E(\Delta y)_{\Delta x}$ is considered as signal noise due to distance change of the probe. Therefore, $f(\varphi)_{i,i+1}$ presents the change of the slope between two probes on the profile terrain. The feature vector V was composed of an $n\times m$ matrix, where the row m represents total probes with n features. In contrast to the conventional HMM, the observed atomic data point was transformed to the numerical feature vector for SVM pattern recognition.

2.4.3 Hidden state assignment In order to combine the observation of sequence Y represented by the feature vector V with the hidden state sequence e, each probe was assigned with one of following labels: expressed or silent state. The assigning criteria were initially based on the NCBI genome annotation data, which serves as a good starting systematic biological knowledge. The sequences were preliminarily labeled as expressed or silent state corresponding to coding and non-coding regions, respectively.

2.4.4 The training set selection To select the training set $\mathcal{X}$, we used a comprehensive and objective scheme to determine subregions from the array data only based on genome annotation data. It has been known that >90% of open reading frames (ORFs) were expressed in *Escherichia coli* (Selinger *et al.*, 2000). Under this assumption, first type of the subregions, which were restricted to the ORF, tRNA and rRNA regions identified by the annotation, was used as training data for learning the expressed state. For learning the silent states, second type of the subregions located at intergenic regions was extracted and 300 bp of the DNA sequence was trimmed from both 5′ and 3′ ends. The trimming was to remove potential ncRNA signals from the untranslated regions (UTRs) both upstream and downstream of ORFs. According to these criteria, 955 regions were selected. Different experimental conditions may influence the selection of the training set. Therefore, leave-one-out cross-validation was used to evaluate each member of the training set for different conditions. Those with accuracy predicted by cross validation lower than 0.9 were considered non-informative or misinformative, and were removed.

2.4.5 The post-processing In the output, every probe was tagged with either expressed or silent state. Therefore, the boundaries of the segments

were defined as the junctions between two regions of different states. In addition, to remove possible false segments, labels of segments consisting fewer than four probes were averted to the same states as the neighbors. In other words, we ignored the small RNA transcripts covered by fewer than four probes (ca. 50 bp in size).

The source code of SVMhmm V3.10 written in C was downloaded from http://www.cs.cornell.edu/People/tj/svm_light/svm_hmm.html (Joachims *et al.*, 2009). The HM-SVMs algorithm was directly implemented in R and the input file for model learning and classification was formatted as described by the program. The results of transcriptome profile, algorithm classification and detail program scripts can be accessed on the website at http://bioinformatics.forsyth.org/mtd.

2.5 Strand-specific reverse transcription polymerase chain reaction

For standard reverse transcription–polymerase chain reaction (RT–PCR), the false positive PCR artifact was detected in the absence of added RT primer due to self-priming of RNA or non-specific small DNA or RNA contaminants in the RNA extraction, which acted as the primers for RT reaction. To avoid the artifacts, a DNA tag non-homologous to *P.gingivalis* W83 genomic sequence was added to the 5′ end of the synthesized RT-primer for reverse transcription. After cDNA was synthesized, regular PCR was performed with the tag and a gene-specific oligo as the primer pair (Purcell *et al.*, 2006). Reverse transcription was initiated with 2 μg RNA and 20 pmol tagged primer. The procedures of RT–PCR followed Invitrogen SuperScript II's manual.

3 RESULTS AND DISCUSSION

3.1 Data normalization and regression

To monitor the signal adjustments during different stages of data processing, we plotted the probe intensities along the genomic coordinate. Raw signal intensities were first adjusted using genomic DNA hybridization data as the reference. Figure 1a shows the genomic DNA intensities on a region of genome. The probe intensities with repeated sequences which may mislead abundance of RNA molecules were regressed to a single copy level. Theoretically, every DNA signal based on single copy number should be of the same level. However, the result showed a much fluctuated intensity profile on the plot because of various factors. One major factor was oligonucleotide composition (Royce *et al.*, 2005), which directly affects the affinity between targets and probes. Another involved the labeling efficiency of the probe. In our study, the genomic DNA was labeled by covalently attaching fluorescent dye to a heteroatom on guanine residues, thus the efficiency of labeling depended on the nucleic acid composition of the probe. The fluctuated DNA hybridization profile was used to adjust the expression profiles so that the bias due to sequence composition can be eliminated.

Figure 1c shows the signal intensities adjusted by DNA reference signals and between-array normalization. Compared to Figure 1b, which plotted original RNA hybridization measurements on a base 2 logarithmic scale, the fluctuation of adjusted intensities was notably reduced within the expressed regions (Fig. 1c), but was amplified in the background regions. This can be confirmed by quantitative comparison of standard deviation of RNA raw data and normalization data in Table 1. Also, the background noise in Figure 1c shows a symmetric and stochastic distribution not similar to Figure 1b. Our explanation is that background intensities are mainly contributed by non-specific noises without perfectly matched RNA target binding. Thus, the background regions are not the suitable targets for the DNA normalization method. Despite of this caveat, two major advantages gained still justified the use of normalization with DNA signals. First, the difference between the positive and the background signals was enhanced, which increased the sensitivity for detecting low level of expression

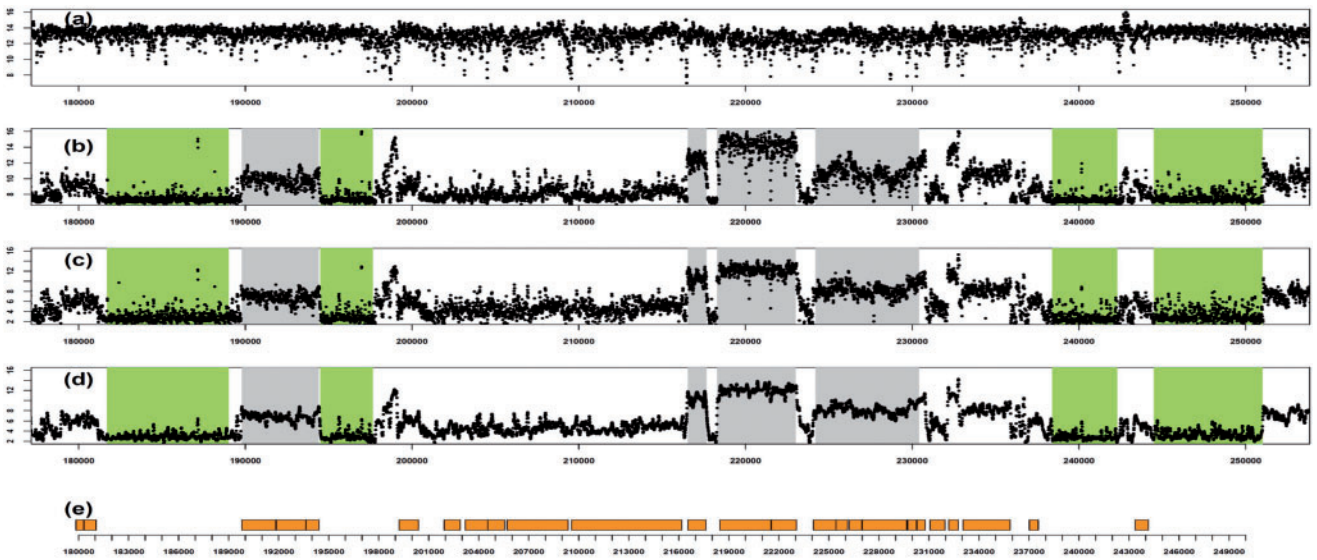

Fig. 1. Intensity profiles of tiling microarray data at various stages of data processing. Log_2 probe intensities (*y*-axis) from a range of 90-kbp of the genome on the sense strand of the sequence were plotted on the genome coordinates (*x*-axis). The plots are (**a**) signal intensities of DNA reference array, (**b**) RNA raw intensities from microarray, (**c**) signal intensities after normalization, (**d**) after adjustment by SVR and (**e**) the corresponding ORFs. The gray areas were selected as the positive signals for further quantitative evaluation of data processing performance as shown in Table 1. The green areas were visually determined as background noise and its quantitative evaluation can be seen in Table 1.

Table 1. Quantitative assessment of different steps of data processing by signal to noise ratio (SNR)[a]

Type	Start (bp)	Stop (bp)	RNA raw intensities			Normalization			SVR		
			Mean	SD	SNR	Mean	SD	SNR	Mean	SD	SNR
Positive	189 800	194 400	9.7	0.66		4.8	0.50		4.8	0.36	
Positive	218 300	223 000	14.1	0.98		7.7	0.47		7.6	0.29	
Positive	216 550	217 650	12.4	0.60		8.1	0.62		8.1	0.47	
Positive	224 200	230 400	10.5	0.93		5.5	0.63		5.5	0.53	
Background	181 700	189 000	7.4	0.34	6.99	2.0	0.67	6.85	2.1	0.37	9.96
Background	194 500	197 650	7.5	0.46		2.1	0.76		2.2	0.60	
Background	238 400	242 300	7.3	0.38		2.0	0.72		2.0	0.38	
Background	244 500	251 000	7.6	0.49		2.2	0.80		2.3	0.48	

[a]SNR was calculated as the following where μ^{Q}_{pos} defines the mean of all positive intensities within 2.5–97.5% quantiles of total, and $\sigma^{Q}_{r\ r\in\text{pos,neg}}$ indicates the mean of the SDs from both positive and negative signals within the same quantiles. $\text{SNR} = (\mu^{Q}_{\text{pos}} - \mu^{Q}_{\text{neg}})/\sigma^{Q}_{r\ r\in\text{pos,neg}}$.

signals. Table 1 shows the difference of mean intensities between positive and background regions prior to and after normalization. Second, the normalized intensities within a transcript expressed a much condensed level and thus better represented the pattern profile of a RNA transcript.

For this reason, SVR was applied to predict the local tendency from the scattered data distribution and eliminated the outliers. In Figure 1d and Table 1, the noises presented by the standard deviations were significantly reduced in all regions after SVR adjustment. The moderation of SD mainly contributed to the improvement of signal-to-noise ratio (SNR), since the difference of positive and background signals did not change after normalization and SVR. To maintain the local tendency of the distribution, we empirically set the parameter σ to 5×10^4, which could fit our highly non-linear data model and reduce the cross-validation error. The smaller the σ was, the much smoother the distribution became. But more informative intensities were lost and cross-validation error increased.

Our major goal was to more accurately differentiate the transcript units (expressed signals) from the background (silent regions). The combination of DNA normalization and SVR enhanced the separation of positive signals from background noises. Figure 2a–c compares the histograms of raw, normalized and SVR transformed intensities. For raw intensities, the majority of transcribed signals are located at levels just above the dominant background noises in accordance with a power-law distribution (Royce *et al.*, 2005). The mixed distribution obscured the recognition of the transcribed regions by the following algorithm. After adjustment by normalization and SVR, the data transformed to a bimodal-like distribution composed of a mixture of two Gaussian distributions. The lower dominant peak comprised the background signals and the right small one represented the transcript signals (Fig. 2b and c). The much-separated distributions between transcript and background signals improved the performance of the machine learning algorithm described next.

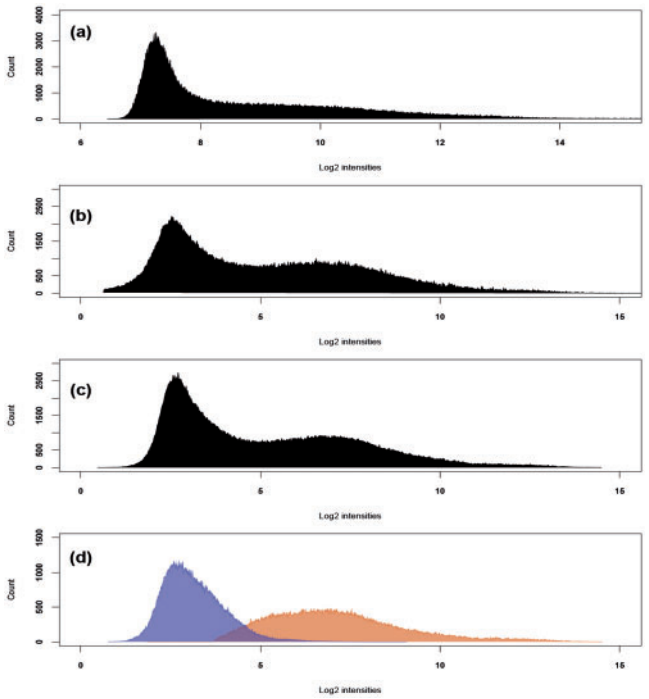

Fig. 2. Histogram of frequency intensity distribution at different stages of data processing. (**a**) RNA raw intensities from microarray; (**b**) RNA intensities after normalization; (**c**) RNA intensities after adjustment by SVR; and (**d**) component distributions combining the background noise distribution (purple area) and positive signal distribution (orange area) from the predicted HM-SVMs results.

3.2 Performance measurement on simulated data

To compare the performance of our HM-SVMs and other methods, the synthetic dataset with four different sizes (10, 20, 30 and 40 K data points) simulating probe intensity distribution along the genomic coordination were constructed. Data points were randomly generated with Gaussian noise by fixing mean and SD (noise). Two hidden states (positive and background) were pre-assigned and the level of positive signals was controlled by SNR.

To optimize the algorithm, several parameters were systematically explored in order to minimize misclassification rate on the simulated dataset, which included the cost of constraints violation C, precision value epsilon ε and the search window size s. Increasing the value of C expanded the cost of misclassified points and forced to create a more accurate model. The value of C was set as 170 in this study.

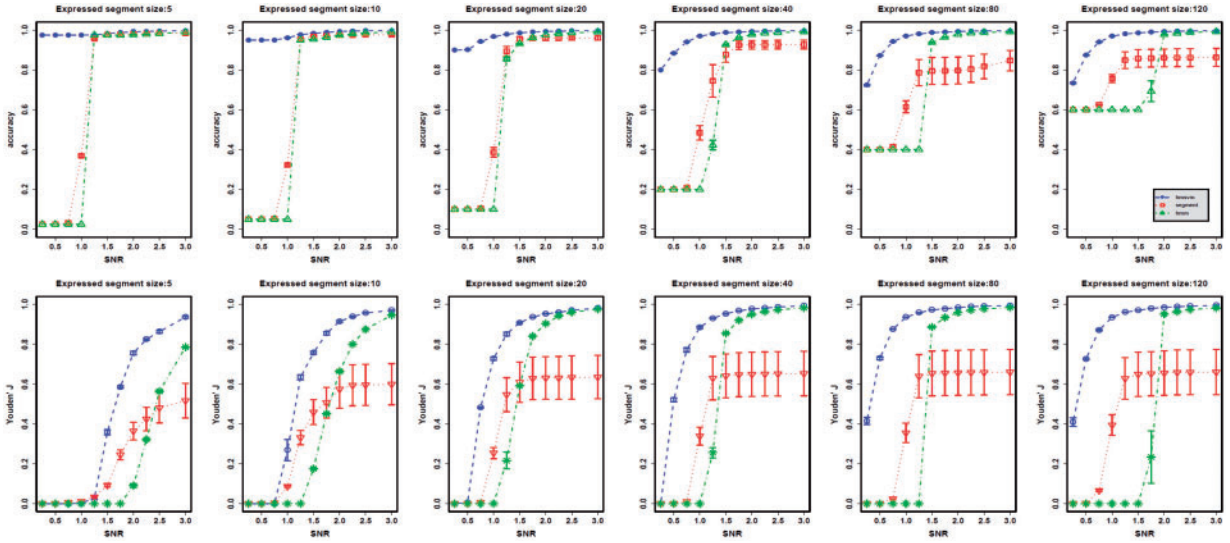

Fig. 3. The performance versus different SNRs and segment sizes among three algorithms—HM-SVMs, HMM and Segmentation (shown in blue, green and red lines, respectively). The bar denotes the 95% confidence level for each data point.

Smaller precision value ε enhanced the prediction accuracy, but the computing time and memory usage increased. In this study ε was set at 0.5. The search window in size S was set at 5. Furthermore, we tested both directions of reading the sequential data points, HM-SVMs generated exactly the same classification with simulated data.

On four different sizes of simulated data, we measured performance of our HM-SVMs and two other methods—the segmentation method by Huber *et al*. (2006) and HMM-based method by Nicolas *et al*. (2009). Since the level of SNR and the size of segment may influence the performance, we challenged the algorithms with various combinations of the two parameters. When the data represent gene expression level, the level of SNR corresponds to the amount of RNA. The size of segment suggests the length of RNA transcript. Some small regulatory RNAs may be found in small segments and operons consisting multiple cotranscribed genes may correspond to large segments. To access the performance of the three algorithms, accuracy and Youden's index (the difference of sensitivity and false positive rate) were calculated with different SNRs and segment sizes (Fig. 3). At low SNR (<1.5), our algorithm outperformed other two methods in different segment sizes. The accuracy of our algorithm was equal or higher than that of other algorithms under all conditions. The accuracy decreased with lower SNR and larger segment size. This may be due to that some data points in segments with larger size and low SNR were classified as background. Comparison of Youden's index, our algorithm showed a better power to discriminate true positive and true negative. However, when SNR was lower (<0.5), all algorithms showed poor Youden's indices, suggesting more false positive predictions. Overall, compared to the other two methods, our HM-SVMs showed higher discriminative power for classifying the underline states under all test conditions.

3.3 Identification of transcription units by HM-SVMs

We applied this algorithm to the analysis of the transcriptome profile for *P.gingivalis* W83. The selected training data for model learning consisted of pairs of input and output objects. The input was the feature vector, based on the distribution of probe intensity expression level described above, and the output was desired hidden states (expressed or silent) retrieved from NCBI genome annotation data. Additionally, any other system-wide experimental data can also be incorporated into the feature vector to enhance the discriminating ability of SVMs. The strategy of training set selection described in the methods was to extract the most informative regions by removing those that may be undergoing post-transcriptional modifications or transcription regulations. The probes in the selected regions were tagged with confident hidden states and were used in the learning of the discriminative function F. In addition, we included leave-one-out cross-validation to evaluate each member of the training set, and the one with non-informative or misinformative was eliminated. This annotation-based selecting strategy was objective but may include false labeling, which can be alleviated by implementing a soft margin created by the error cost C and the slack variable ξ in the discriminative function F.

The distributions of probe intensities associated with both expressed and silent states from HM-SVMs results were plotted in Figure 2d. Degrees of separation of these two distributions showed that HM-SVMs were able to recognize the patterns from both states and to discriminate them correctly. Notably, there was an overlap between the two distributions, suggesting that the decision of classification was determined by corresponding feature vector, which was highly related to local tendency (subsequent dependency), instead of by a simple cutoff. Therefore, the algorithm could recognize the complex pattern of the expression profile and make a decision on the expression state intelligently.

Examples of transcription profile analysis using HM-SVMs were shown in Figure 4. The figure shows that the algorithm was able to correctly distinguish the expressed transcripts from background noises consistent to the annotated genes and intergenic regions, respectively. Several types of RNA transcripts were classified by comparing the HM-SVMs results with the genome annotation. Examples of different types of RNAs were shown in Figure 4.

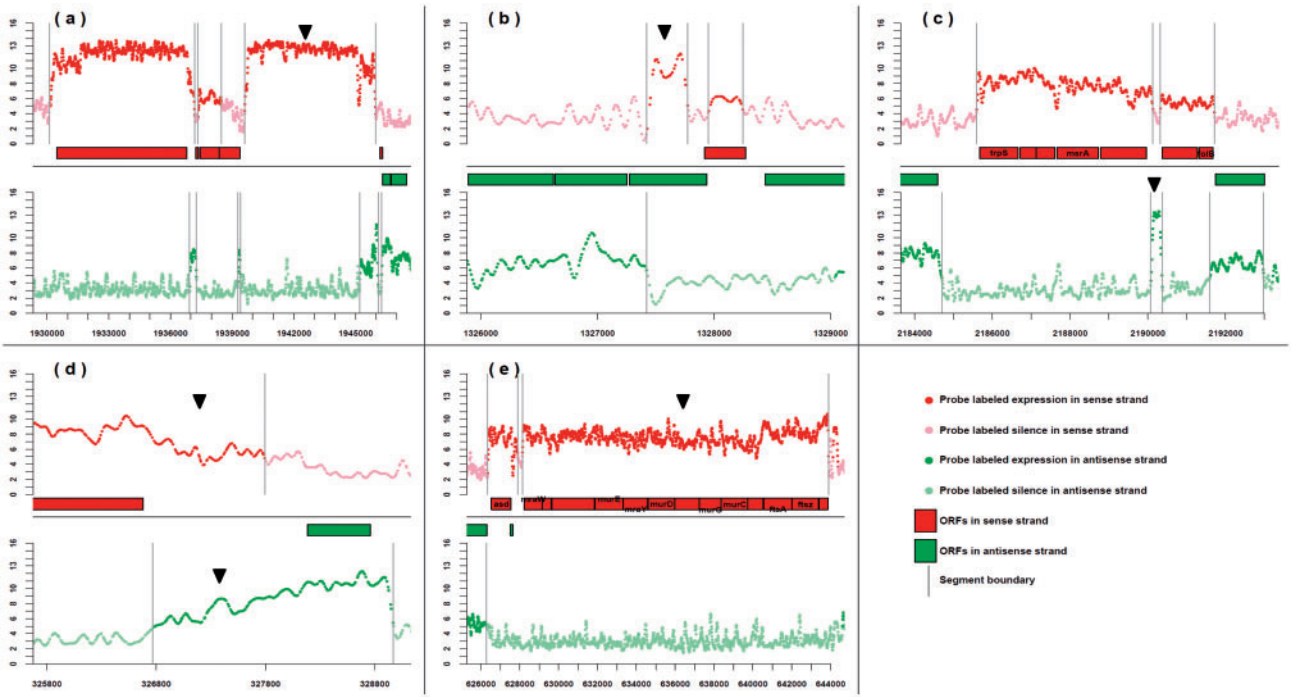

Fig. 4. Examples of transcription prediction made by HM-SVMs. Normalized log intensities (*y*-axis) of the probes were plotted against the actual probe coordination of the genome (*x*-axis). The data predicted with the 'expressed' states were colored in red and green on the sense and antisense genomic sequences, respectively; data with 'silent' states were colored in pink and light green on the sense and antisense, respectively. Computer predicted ORFs were shown in red and green boxes on the sense and antisense strand, respectively. Putative boundaries of the transcripts were indicated as vertical gray lines. Several different types of RNA transcripts (exemplified by arrowheads) categorized are as follows: (**a**) novel RNA transcripts, (**b**) antisense RNAs, (**c**) non-coding small RNAs, (**d**) putative 5′ UTR region and (**e**) potential operon containing multiple ORFs.

We found a large number of transcripts containing potential novel ORFs in not yet annotated regions (Fig. 4a). Several *cis*-encoded antisense RNAs (Brantl, 2007) opposite to the location of the sense transcripts were also identified in many regions (Fig. 4b). Small ncRNAs which may be responsible for regulation of its antisense gene expression were also observed (Fig. 4c). The 5′ and 3′ UTR regions of the transcripts (Fig. 4d) may provide useful resource for studying post-transcriptional regulation. Furthermore, large transcription units were frequently found to contain several ORFs (Fig. 4e) and are the typical operons transcribed from the same promoters in bacterial cells.

3.4 Benchmark comparison

The HM-SVMs algorithm used in this study is highly efficient in terms of computational time. The analysis of the single strand of *P.gingivalis* genomic tiling array expression profile (ca. 200 K data points) required less than 1 min of computation time and consumed only 120 Mb system memory on a single core 2.3 MHz Intel-based computer. For the same analysis on the same computer platform, the segmentation algorithm published by Huber *et al.* (2006) took more than 16 h [parameters: maximum segment length $l = 2500$ (25 kbp) and maximum segment number $K = 1900$] and the method implementing HMM framework (Nicolas *et al.*, 2009) took more than 6 h (with the parameter hidden state K set at 100). Clearly, the HM-SVMs based algorithm reported here is much more efficient compared to other algorithms used for transcriptome profile analysis.

3.5 Predicted transcripts validated by RT–PCR

To validate the novel transcripts and transcriptional architectures by the HM-SVMs algorithm, a total of 36 regions predicted by HM-SVMs as either expression or silence were subjected to experimental verification by RT–PCR. Of 15 selected expressed regions by HM-SVMs, 13 showed positive RT–PCR signals, indicating the presence of RNA. Of 21 selected silent regions, 18 showed no sign of RT–PCR product, thus confirming the lack of RNA in these regions. By calculating hypergeometric distribution probability of classification from both experiment and computer, it suggested that the prediction of the algorithm significantly matched the results of the experiment (P-value <0.05). Moreover 7 of total 15 predicted expressed regions confirmed by RT–PCR are novel RNAs that have never been described before. Of three regions which were predicted as silent but showed positive RT–PCR signals, the intensities in these regions were very close to the background noises (SNR ≈ 0.01) and thus the expression patterns were masked or disrupted by the background. Localized and better background correction algorithm may be needed in order to increase the accuracy for the prediction of the low intensity area.

4 CONCLUSIONS

A comprehensive method combining multiple innovative algorithms was devised and used for transcriptome profile analysis. This method starts with raw data normalization using DNA reference array to

estimate probe-specific scaling and background parameters and to adjust probe intensity accordingly. SVR algorithm, a regression model preserving the local tendency of the profile, was then used to minimize the noises caused by the stochastic measurement error. Normalized and smoothed profile data were then subjected to expression status prediction using heterogeneous HM-SVMs, which incorporates profile geometry learning and transforms the hybridization signals into the high-dimensional feature vector for the discriminative model training. Viterbi algorithm was then used to decode the most likely underlying sequential states, considering the dependencies on the neighboring states. The performance was evaluated with the simulated data and our HM-SVM outperformed two other algorithms designed for similar purpose. The HM-SVMs algorithm has the flexibility of combining different types of validated biological information in learning the feature vector for constructing the predicting function. We believe this method will be a great addition to the current methods available for transcriptome profile analysis. Furthermore, this method can also be applied to the analysis of other types of time serial data, such as CGH, ChIP-on-chip and RNA-sequencing data for predicting hidden states of the data points.

ACKNOWLEDGEMENTS

We thank Dr Mark Kon at Boston University for valuable comments on the writing of the manuscript.

Funding: National Institute for Dental and Craniofacial Research (grant no. R21 DE018803-01A1); Mobility grant from the Faculty of Dentistry, University of Oslo, Oslo, Norway (to H.H.).

Conflict of Interest: none declared.

REFERENCES

Akama,T. *et al.* (2009) Whole-genome tiling array analysis of Mycobacterium leprae RNA reveals high expression of pseudogenes and noncoding regions. *J. Bacteriol.*, **191**, 3321–3327.

Altun,Y. *et al.* (2003) Hidden Markov support vector machines. In *Proceedings of the Twentieth International Conference on Machine Learning*. Washington, DC, pp. 3–10.

Bai,J. and Perron,P. (2003) Computation and analysis of multiple structural change models. *J. Appl. Econometrics*, **18**, 1–22.

Bertone,P. *et al.* (2004) Global identification of human transcribed sequences with genome tiling arrays. *Science*, **306**, 2242–2246.

Brantl,S. (2007) Regulatory mechanisms employed by cis-encoded antisense RNAs. *Curr. Opin. Microbiol.*, **10**, 102–109.

Cortes,C. and Vapnik,V. (1995) Support-vector networks. *Mach. Learn.*, **20**, 273–297.

David,L. *et al.* (2006) A high-resolution map of transcription in the yeast genome. *Proc. Natl Acad. Sci. USA*, **103**, 5320–5325.

Du,J. *et al.* (2006) A supervised hidden Markov model framework for efficiently segmenting tiling array data in transcriptional and chIP-chip experiments: systematically incorporating validated biological knowledge. *Bioinformatics*, **22**, 3016–3024.

Duncan,M.J. *et al.* (1993) Interactions of Porphyromonas gingivalis with epithelial cells. *Infect. Immun.*, **61**, 2260–2265.

Furey,T.S. *et al.* (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, **16**, 906–914.

Hovik,H. and Chen,T. (2010) Dynamic probe selection for studying microbial transcriptome with high-density genomic tiling microarrays. *BMC Bioinformatics*, **11**, 82.

Huang,K. *et al.* (2006) Local support vector regression for financial time series prediction. *International Joint Conference on Neural Networks*. Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, pp. 1622–1627.

Huber,W. *et al.* (2002) Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, **18** (Suppl. 1), S96–S104.

Huber,W. *et al.* (2006) Transcript mapping with high-density oligonucleotide tiling arrays. *Bioinformatics*, **22**, 1963–1970.

Joachims,T. *et al.* (2009) Cutting-plane training of structural SVMs. *Mach. Learn.*, **77**, 27–59.

Kapranov,P. *et al.* (2002) Large-scale transcriptional activity in chromosomes 21 and 22. *Science*, **296**, 916–919.

Karatzoglou,A. *et al.* (2004) Kernlab–an S4 package for kernel methods in R. *J. Stat. Software*, **11**, 1–20.

Li,W. *et al.* (2005) A hidden Markov model for analyzing ChIP-chip experiments on genome tiling arrays and its application to p53 binding sequences. *Bioinformatics*, **21** (Suppl. 1), i274–i282.

Li,L. *et al.* (2007) Global identification and characterization of transcriptionally active regions in the rice genome. *PLoS One*, **2**, e294.

Munch,K. *et al.* (2006) A hidden Markov model approach for determining expression from genomic tiling micro arrays. *BMC Bioinformatics*, **7**, 239.

Nicolas,P. *et al.* (2009) Transcriptional landscape estimation from tiling array data using a model of signal shift and drift. *Bioinformatics*, **25**, 2341–2347.

Picard,F. *et al.* (2005) A statistical approach for array CGH data analysis. *BMC Bioinformatics*, **6**, 27.

Purcell,M.K. *et al.* (2006) Strand-specific, real-time RT-PCR assays for quantification of genomic and positive-sense RNAs of the fish rhabdovirus, Infectious hematopoietic necrosis virus. *J. Virol. Methods*, **132**, 18–24.

Rabiner,L.R. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77**, 257–286.

Royce,T.E. *et al.* (2005) Issues in the analysis of oligonucleotide tiling microarrays for transcript mapping. *Trends Genet.*, **21**, 466–475.

Schadt,E.E. *et al.* (2004) A comprehensive transcript index of the human genome generated using microarrays and computational approaches. *Genome Biol.*, **5**, R73.

Selinger,D.W. *et al.* (2000) RNA expression analysis using a 30 base pair resolution Escherichia coli genome array. *Nat. Biotechnol.*, **18**, 1262–1268.

Stolc,V. *et al.* (2004) A gene expression map for the euchromatic genome of Drosophila melanogaster. *Science*, **306**, 655–660.

Tjaden,B. *et al.* (2002) Transcriptome analysis of Escherichia coli using high-density oligonucleotide probe arrays. *Nucleic Acids Res.*, **30**, 3732–3738.

Yamada,K. *et al.* (2003) Empirical analysis of transcriptional activity in the Arabidopsis genome. *Science*, **302**, 842–846.

Zeller,G. *et al.* (2008) Transcript normalization and segmentation of tiling array data. *Pac. Symp. Biocomput.*, 527–538.