Integrating copy number polymorphisms into array CGH analysis using a robust HMM [*]

Sohrab P. Shah [a], Xiang Xuan [a], Ron DeLeeuw [b], Mehrnoush Khojasteh [b],
Wan Lam [b], Raymond Ng [a], Kevin P. Murphy [a]

a - Department of Computer Science, University of British Columbia, 201-2366 Main Mall Vancouver, BC V6T 1Z4 Canada

b - British Columbia Cancer Research Centre, 675 West 10th Avenue Vancouver, BC V5Z 1L3 Canada

Abstract

Array comparative genomic hybridization (aCGH) is a pervasive technique used to identify chromosomal aberrations in human diseases, including cancer. Aberrations are defined as regions of increased or decreased DNA copy number, relative to a normal sample. Accurately identifying the locations of these aberrations has many important medical applications. Unfortunately, the observed copy number changes are often corrupted by various sources of noise, making the boundaries hard to detect. One popular current technique uses hidden Markov models (HMMs) to segment the signal into regions of constant copy number; a subsequent classification phase labels each region as a gain, a loss or neutral. Unfortunately, standard HMMs are sensitive to outliers, causing over-segmentation. We propose a simple modification that makes the HMM more robust to such single clone outliers. More importantly, this modification allows us to exploit prior knowledge about the likely location of such "outliers", which are often due to copy number polymorphisms (CNPs). By "explaining away" these outliers, we can focus attention on more interesting aberrated regions. We show significant improvements over the current state of the art technique (DNAcopy with MergeLevels) on some previously used synthetic data, augmented with outliers. We also show modest gains on the well-studied H526 lung cancer cell line data, and argue why we expect more substantial gains on other data sets in the future. Source code written in Matlab is available from `http://www.cs.ubc.ca/~sshah/acgh` Contact: `sshah@cs.ubc.ca`.

1 Introduction

Array comparative genomic hybridization (aCGH) is a high-throughput cytogenetic technique to measure DNA copy number changes in a disease sample compared to a normal sample [19]. Chromosomal aberrations that exhibit DNA copy number changes are indicative of numerous diseases including cancer and mental retardation. Identifying such aberrations can help to locate diagnostically important regions in the genome, which harbour differentially expressed genes. Application of aCGH is widespread in molecular analysis of cancer and holds great promise as a technique to identify clinically relevant diagnostic biomarkers.

The aCGH technique is based on spotting clones that span a discrete region in the human genome on an array. The size and number of clones vary depending on the technological platform and the desired resolution: see Pinkel and Albertson [19] for a review. In this paper, we use data from four chromosomes from H526 (a well-studied cancer cell line) generated using sub-megabase resolution tiling arrays (SMRT) [13]. The output of all such methods is represented as a $\log_2$ ratio of the reference and tumour fluorescence intensities, which are proportional to copy numbers. So in a neutral state, one would expect to see $\log_2(2/2) = 0$; with one copy lost, one would expect to see $\log_2(1/2) = -1$; with one gain $\log_2(3/2) = 1.58$, etc. The goal of analysis techniques is to detect the regions of changed copy number (i.e., to segment the signal), and then to label each region as loss, neutral or gain (sometimes it is useful to distinguish gains of 1 copy from gains of more than 1); we call this latter task "classification".

In reality, the observed data is much more complex than the above description suggests. Figure 5 (A) shows a typical plot of aCGH data for one chromosome from H526 (see Section 3.2 for more details on H526). The red squares represent copy number losses and the green circles represent regions of gain. (In this example, the aberrated regions were identified manually by an expert cytogenetecist.) The figure demonstrates that although copy number changes in DNA is a theoretically discrete process, the intensity ratios for aCGH do not produce a clean piecewise constant signal. Also note that aberrated regions tend to span contiguous sets of clones along a chromosome, although some aberrations can be as small as a single clone. This suggests that any analysis technique should exploit such spatial correlation.

In Figure 5(A), we also depict 'outlying' clones (detected by eye) with light blue triangles. Treating such points as inliers can significantly affect the remaining points, by causing over-segmentation, for example. There are several possible causes of such outliers. The first is that they truly represent aberrated regions. The second is some kind of measurement

[*] This is an extended version of a paper submitted to ISMB'06.

noise, or mislabeling (sometimes the locations of clones is mis-recorded). Finally, there is the possibility that the single clone outliers correspond to known locations of copy number polymorphisms (CNPs). Examples of CNPs are shown as dark blue diamonds in Figure 5(A).

The full impact of CNPs on aCGH analysis is not yet known, however indications from two recent large scale studies by Sebat *et al* [24] and Iafrate *et al* [12] measuring background frequencies of copy number variations in the normal human population have revealed hundreds of loci in the genome that are polymorphic in copy number. Buckley *et al* [2] suggest that the results produced by these two studies represent the "tip of the CNP iceberg". For example Sebat *et al* report a CNP at a gene involved in food intake, suggesting a differential propensity for obesity. They also report CNPs at loci related to neurological development and at loci implicated in leukemia and breast cancer drug resistance [24]. These latter examples indicate that for cancer studies, the 'baseline' copy number should be considered when assessing aberrations. We anticipate that the impact of CNPs will be greater on high-resolution arrays and/or full genome coverage arrays, as they are intended to reveal all aberrations in a sample and will detect a larger number of CNPs.

1.1 Our contribution

In this paper, we introduce a way of extending the HMM framework proposed in Guha et al. [9] to handle outliers and CNPs. The basic idea is to replace the Gaussian observation model with a mixture of Gaussians; one mixture component represents the $\log_2$ ratio we would expect from the given state (loss, neutral or gain); the other mixture component represents the $\log_2$ ratio we would expect from an outlier. This simple change makes the model much more robust.

More significantly, we can incorporate knowledge about CNPs into the mixing weights of the mixture model. That is, we can set the prior probability of using the outlier component at location i to the known frequency of CNPs at location i, if i overlaps with a known CNP location; otherwise we set it to the general background outlier probability (which is estimated from data). We explain our model in more detail in Section 2.1.

Several authors (e.g., [9, 23]) propose estimating the parameters of the HMM using MCMC (Markov chain Monte Carlo) techniques, as opposed to the more common EM (expectation maximization) algorithm. The advantage of MCMC is that it provides full posterior estimates over the parameters, rather than just point estimates, thus properly modeling uncertainty (see e.g., [8] for an introduction to MCMC and Bayesian data modeling). MCMC also partly mitigates problems with local minima than EM is well known to suffer from. It also turns out to be simpler to exploit informative prior constraints in a sampling framework than in an optimization framework. We explain how to perform efficient MCMC in Section 2.2.

We first evaluate performance of our model on a synthetic dataset published in Willenbrock and Fridlyand [25]. The advantage of using synthetic data is that the true locations of the aberrations are known, so we can assess performance reliably. In addition, we can control the difficulty of the problem. The Willenbrock data is considerably harder (but more realistic) than other synthetic datasets used in earlier papers. We make the Willenbrock data even harder by adding outliers, to check the robustness of our method. We compare our method to DNAcopy+MergeLevels (using default parameters), which has been shown in two previous comparative studies [25, 16] to be a leading current method. Henceforth we will refer to this method as MergeLevels. Having established that our method is better than current techniques on synthetic data, we then applied it to real H526 data. Our results are in Section 3, which we discuss in Section 4.

1.2 Related work

A recent survey paper by Lai *et al* [16] describes and evaluates eleven algorithms for aCGH data analysis. We can loosely group these methods into three main approaches: smoothing, segmentation, and combined segmentation and classification. Smoothing approaches such as Quantreg, developed by Eilers and Menezes [5], and the wavelet approach of Hsu *et al* [10], attempt to fit a curve to the data, while handling abrupt changes. Smoothing methods generally filter the data using a fixed size window, and therefore will be unable to detect outliers or CNPs that span a single clone. In addition, they are primarily designed as a visual aid interpret the data and do not accomplish the main objective of automatically identifying aberrated clones.

Segmentation methods identify contiguous sets of clones (segments) that share the same mean $\log_2$ ratio. The output of the segmentation methods usually consists of the boundaries and means of the segments. The clones within a segment are assumed to share the same copy number. We refer to the boundaries of segments as breakpoints. Examples of segmentation algorithms include DNACopy [17], which is based on a recursive circular binary segmentation algorithm; CGH-Seg [18] which uses a penalised likelihood model to determine breakpoints; aCGH-Smooth [14], which uses a genetic algorithm to find breakpoints; and the GLAD method of Hupe *et al* [11], which includes a median absolute deviation model to explicitly treat outliers as separate from its surrounding segment. In Lai's comparison, CGHSeg and DNACopy are consistently the best. Willenbrock and Fridlyand [25] compared performance of DNACopy and GLAD and report better performance with DNACopy. We therefore use DNAcopy as our baseline model.

A general limitation of segmentation is that the output needs to be further analysed in order to infer which segments are aberrated regions, i.e., to "call" the gains and losses. Methods such as GLADMerge [11] and MergeLevels [25] perform this post-processing task by merging together segments with "similar" mean levels, and then classifying them. However, as noted by Engler *et al* [6] and Willenbrock and Fridlyand [25], it is much better to perform the segmentation and classification simultaneously, since the class labels can help with the segmentation as well as vice versa.

An obvious way to perform simultaneous segmentation and classification is to use an HMM. The first approach to do this was by Fridyland et al [7]. However, in their approach, the states of the HMM do not have any intrinsic meaning (they are just indices to represent a discrete number of mean levels, typically $K = 5$). Hence post-processing was necessary to come up with labels. Guha et al. [9] modify this to use a "supervised" 4-state HMM, where the states are defined to mean loss, neutral, one-gain or multiple-gain. The advantage of this is two-fold: first, it is easy to perform simultaneous segmentation *and classification* using the Viterbi algorithm; secondly, we can impose informative priors on the parameters, since they now have biological meaning. This paper extends the 4-state HMM model by adding robustness to outliers and location-specific priors (LSPs), which can be used to encode CNPs.

In addition to the work mentioned above, two recent papers have explored some interesting variations. Broet and Richardson [1] propose using a latent 1D Gaussian random field, as opposed to a latent 1D discrete random field (i.e., an HMM), to model spatial correlation between levels. However, this does not solve the classification problem. Engler *et al* [6] introduce spatial dependence by breaking the data into overlapping triples, and then using a hierarchical random effects model. Unfortunately, because the triples are overlapping, the data is overcounted, so optimizing the likelihood turns out to be intractable. Instead, they compute a local maximum of the pseudolikelihood. We also use a hierarchical Bayesian model, but we are able to compute posterior estimates using an exact likelihood function.

2 Methods

2.1 Our model

Our basic model very similar to the 4-state HMM in Guha et al [9], where the states represent loss, neutral, one-gain and multiple-gain. (We also tried a 3-state model, where we combined all the gain states, but results were not as good.) The main difference from Guha is that the observation density is a mixture of 2 Gaussians, one representing inlier (clones belonging to one of the loss, neutral or gain states) and the other representing outlier. We introduce binary indicator variables $O_i \in \{0, 1\}$ where $O_i = 1$ means location i is an outlier, and $O_i = 0$ means it's an inlier. Then the class-conditional density becomes

$$p(y_i|O_i, S_i = s) = \begin{cases} \mathcal{N}(y_i|\mu_0, \sigma_0) & \text{if } O_i = 1 \\ \mathcal{N}(y_i|\mu_s, \sigma_s) & \text{if } O_i = 0 \end{cases} \quad (1)$$

where y_i is the $\log_2$ ratio for clone i where the clones are ordered by their physical location on a chromosome. S_i is the state label at position i. Thus O_i acts like a "switching parent" variable, which selects between the outlier parameters μ_0, σ_0 or the inlier parameters, μ_s, σ_s. The O_i variables are modeled as conditionally independent. Hence, there are no Markovian dynamics on the outliers. This allows the model to make temporary "excursions" to the outlier state, without incurring any "penalty" implicitly encoded by the state transition matrix. Our model is summarized in Figure 1.

Note that we model each chromosome of each sample independently. This is a deliberate design decision, since we do not believe it is reasonable to share parameters (or "borrow statistical strength") across chromosomes or samples, since the data have such different levels (magnitudes) in each sample. Clearly multiple samples of the same chromosome have something in common — this is precisely what scientists hope to discover! However, we do not believe that what they have in common is the same mean levels for each state; rather, it is presumably something more abstract, such as information about *locations* of breakpoints. We plan to pursue this in future work, but for now, we limit our attention to modeling samples separately.

2.1.1 Priors

The parameters of the model are as follows. For each state $j \in \{1, \ldots, 4\}$, we have the mean and variance of the Gaussian, μ_j, σ_j^2. We also have μ_0, σ_0^2 for the outlier state. For each location that is known to be a CNP, we have an outlier probability, $\rho_i = P(O_i = 1)$; for all other locations, we have the "background" outlier probability, ρ_0. Finally, we have the transition matrix A and the initial state distribution π.

We use standard conjugate priors (see e.g., [8]) for all the parameters, as follows:

$$\begin{aligned} p(\mu_s|\sigma_s^2) &= \mathcal{N}(\mu_s|m_s, \frac{\sigma_s^2}{\kappa_s}) && (2) \\ p(\sigma_s^2) &= \chi^{-2}(\nu_s, \tau_s^2) = IG(\frac{\nu_s}{2}, \frac{\nu_s\tau_s^2}{2}) && (3) \\ p(A) &= \prod_{s=1}^{K} Dir(A_{s,\cdot}|\delta_1^A, \ldots, \delta_K^A) && (4) \\ p(\pi) &= Dir(\pi|\delta_1^\pi, \ldots, \delta_K^\pi) && (5) \\ p(\rho_i) &= Beta(a_i, b_i) && (6) \end{aligned}$$

where K is the number of states, $\mathcal{N}$ is a Gaussian, χ^{-2} is an inverse-chi-squared distribution, IG is an inverse Gamma distribution, Dir is a Dirichlet distribution, and Beta is the beta distribution. As is apparent, these priors themselves have parameters, called hyperparameters. These all have intuitive interpretations. m_s is our prior belief about μ_s (the mean of state s), and κ_s is how strongly we believe this (the effective sample size of the prior); τ_s^2 is our prior belief about σ_s^2 (the variance of state s). and ν_s is how strongly we believe this; the

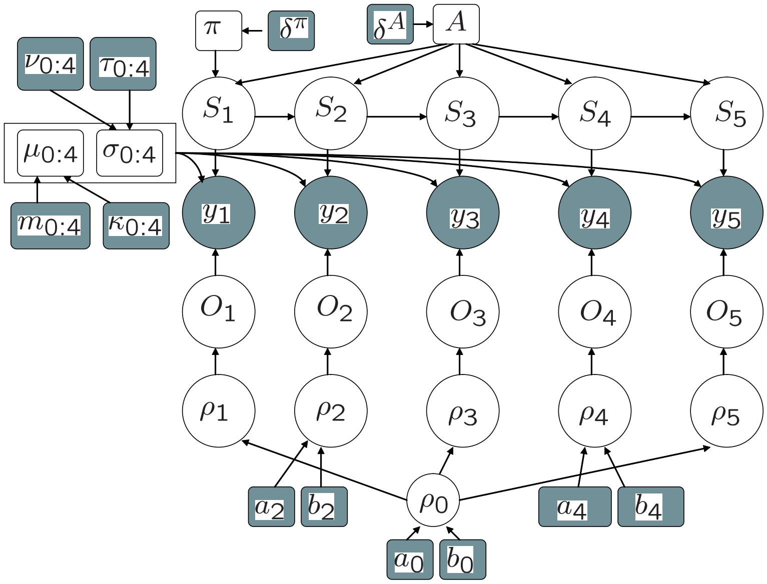

Figure 1: Our model represented as a Bayesian network for a toy chromosome with 5 clones. Square nodes are parameters, round nodes are random variables. Shaded nodes are observed (known), unshaded nodes are hidden (unknown). $S_{1:5}$ represent the states at positions 1 to 5 along the chromosome; $y_{1:5}$ are the observations ($\log_2$ ratio); $O_{1:5}$ indicates if the clone is an outlier or not; $\mu_{1:4}$ and $\sigma_{1:4}$ are the means and variances of states 1 to 4; μ_0 and σ_0 is the mean and variance of the outlier state; $\rho_{1:5}$ are the probabilities of outlier at locations 1 to 5, ρ_0 is the general background outlier probability; π is the initial distribution of states; δ^π are the hyperparameters for π; A is the Markov chain transition matrix; m_j, τ_j are hyperparameters for μ_j; α_j, β_j are hyperparameters for σ_j; a_i, b_i are hyperparameters for ρ_i; δ^A are the hyperparameters for A. Hyper-parameters are shown shaded since they must be set by the user. In this example, we have assumed that locations 2 and 4 correspond to known CNPs; other locations use the background outlier probability ρ_0. Hence $\rho_1 = \rho_3 = \rho_5 = \rho_0$ are all the same.

Param	Loss	Neutral	One-gain	Many-Gain
m_s	-0.1	0	0.57	1
κ_s	0.1	0.1	0.1	0.1
τ_s^2	0.1	**0.01**	0.2	0.2
ν_s	0.1	0.1	0.1	0.1

Table 1: Setting of hyper-parameters for observation model. For the H526 data, we reduced the variance of the neutral state (bold) to $\tau_2^2 = 0.01$, since the data has been median filtered and hence is less noisy than the synthetic data.

	Synthetic		H526	
Param	Outlier	CNP	Outlier	CNP
a_i	0.75	0.5	0.25	f_i
b_i	0.75	0.5	0.25	$1 - f_i$
$E\rho_i$	0.5	0.5	0.5	f_i
Strength	1.5	1	0.5	1

Table 2: Setting of hyper-parameters for outlier process. Here f_i is the population frequency of a polymorphism at location i. We show the parameters and the expected probability of outlier this implies; we also show the strenght of the prior in terms of its equivalent sample size. For the CNPs, we only observe one data point, so we must set the prior carefully. For the other outliers, there is usually enough of them that they overwhelm the prior, so the posterior mean $E[\rho_0|D]$ represents the overall outlier probability (excluding known CNPs).

Bound	Loss	Neutral	One-gain	Many-Gain
Lower	$-\infty$	$-\epsilon/2$	ϵ	$\mu_3 + 3\sigma_3$
Upper	$-\epsilon$	$\epsilon/2$	0.58	∞

Table 3: Setting of the truncation intervals for the means. We require than μ_4 be at least 3 standard deviations above μ_3, so that single gains do not get called as multiple gains.

Bound	Loss	Neutral	One-gain	Many-Gain
Upper	0.41	0.41	0.41	∞

Table 4: Setting of the truncation intervals for the standard deviations for each state, σ_s. By prevent the variance getting too large, we ensure that the "bands" defining each state do not overlap, and that extreme points are explained by the outlier process (or by the many-gain state). The lower bound is 0 since a Gamma distribution is only defined on positive variables.

δ_i parameters of the Dirichlet distributions can be interpreted as pseudo counts; finally, $a_i/(a_i + b_i)$ is the probability of being an outlier at location i, which we believe with strength $a_i + b_i$.

We use a uniform (uninformative) prior for the transition matrix A and the initial state distribution π (i.e., we use $\delta_i^A = \delta_i^\pi = 1/K$), since the signal can start in any state and move from any state to any other state. The remaining parameters are given weakly informative priors, as shown in Table 1 and 2. These values were chosen by hand by looking at the data. More rigorous approaches, based on empirical Bayes [3] or hierarchical priors, could be used.

In order to ensure the model is identifiable (i.e., to avoid label switching), we enforce the following constraint on the mean parameters: $\mu_1 < \mu_2 < \mu_3 < \mu_4$, where the states represent loss, neutral, one-gain and multiple-gain. We do this using a truncated Gaussian prior as follows:

$$\mu_s|\sigma_s^2 \quad \sim \quad \mathcal{N}(\mu_s|m_s, \frac{\sigma_s^2}{\kappa_s})I_s(\mu_s) \qquad (7)$$

where $I_s(\mu_s) = 1$ is μ_s is in interval I_s and 0 otherwise. We use the lower and upper bounds shown in Figure 3, based on [9]; we use $\epsilon = 0.1$.

We define the prior so that $\sigma_{3,4} > \sigma_1 > \sigma_2$, which means that the gain states have higher variance than the loss state, which has higher variance than the neutral state (an empirical fact about most aCGH data). We could also enforce this ordering using a truncated Inverse Gamma prior as follows

$$\sigma_s^2 \quad \sim \quad IG(\frac{\nu_s}{2}, \frac{\nu_s \tau_s^2}{2})I_s(\sigma_s^2) \qquad (8)$$

We use the bounds shown in Table 4, based on [9].

Note that handling truncated priors in EM (for MAP estimation) is tricky, since it would require constrained optimization methods in the M step. However, when using MCMC, there are standard methods for sampling from truncated Gaussians [21] and trunctated Gammas [4].

Prior knowledge about CNPs is encoded as follows. Locations $i \in \mathcal{P}$ which are known to come from CNPs get an adjustable parameter ρ_i which reflects the probability of outlier at that location. The parameters of the (beta) prior on ρ_i is set so that the expected value of ρ_i is equal to the frequency of polymorphisms at that location in the population. Locations $i \notin P$, which are not known to come from CNPs, share the same parameter ρ_0, which represents the background probability of outlier. The (beta) prior on ρ_0 is set so that the expected value of ρ_0 is equal to the expected fraction of outliers. The exact value is not important (we currently use $E\rho_0 = 0.5$) since we use a weak prior strength and we have enough data that the prior is irrelevant. We will let $C = |P|$ be the number of CNP locations, so ρ is a vector of length $C + 1$.

2.2 Algorithm

The output of the algorithm is the following: estimates of the states $\gamma_i(s) = p(S_i = s|y_{1:n})$ and outlier probabilities $\omega_i(o) = p(O_i = o|y_{1:n})$, as well as estimates of the parameters, $p(\theta|y_{1:n})$. We use an MCMC algorithm called block Gibbs sampling to infer these quantities. The key to making this efficient is to use the forwards-filtering backwards-sampling algorithm for HMMs [23]. This is very similar to the more familiar forwards-backwards and Viterbi algorithms,

```
initialize parameters sensibly (eg set to prior mean)
initialize $o^1_{1:n}$ sensibly (eg set $O_i = 1$ if obviously an outlier)
for each iteration $k$
    Compute local evidence $B^k_i(s) = p(y_i|S_i = s, o^t_i, \mu^k, \sigma^k)$
    using Equation 1
    Block $B_1$: sample $s^{t+1}_{1:n}|y, A^k, B^k$ with forwards-backwards
    Block $B_2$: sample $o^{t+1}_{1:n}|y, s^{t+1}, \rho^k, \mu^k, \sigma^k$ in parallel
    Block $B_3$: sample $\mu^{t+1}_{0:4}, \sigma^{t+1}_{0:4}|y, s^{t+1}_{1:n}, o^{t+1}_{1:n}$
    Block $B_4$: sample $\rho^{t+1}_{0:C}|o^{t+1}_{1:n}$ in parallel
    Block $B_5$: sample $A^{t+1}|s^{t+1}_{2:n}$
    Block $B_6$: sample $\pi^{t+1}|s^{t+1}_1$
next k
Compute Rao-Blackwellised estimates:
$\hat{\gamma}_i(s) = \frac{1}{N}\sum_{t=burnin}^{niter} \gamma^k_i(s)$
$\hat{\omega}_i(s) = \frac{1}{N}\sum_{t=burnin}^{niter} \omega^k_i(s)$
```

Figure 2: Pseudo code for the algorithm.

except we sample state sequences from their posterior, rather than computing the most probable sequence or marginal state probabilities. Conditioned on knowing the states, it is easy to update the parameters of the model. The same intuition is used in EM, but the advantage of sampling is that we can model uncertainty in the parameters more easily.

The algorithm is sketched in Figure 2. The running time is $O(NT)$ where $N \sim 1000$ is the number of clones in the input and $T \sim 100$ is the number of MCMC iterations needed to obtain convergence (which we assess informally by monitoring quantities of interest by eye). We explain the steps in more detail below.

2.2.1 Updating $S_{1:n}$

We use the forwards filtering, backwards sampling algorithm for HMMs. This samples paths from the posterior

$$s^*_{1:T} \sim p(s_{1:T}|x_{1:T}) \tag{9}$$

This can be done as follows. First define $B_t(j) = p(y_t|S_t = j, o_t)$ using Equation 1 with the current parameters μ, σ^2, and the current outlier status $o_{1:n}$. Then compute the filtered distributions $\alpha_t(j) = p(S_t = j|x_{1:t})$ and $\xi_{t|t}(i,j) = p(S_{t-1} = i, S_t = j|x_{1:t})$ using the standard forwards equations (see e.g., [20]).

$$\alpha_1 \quad \propto \quad B_1\pi \tag{10}$$

$$\alpha_t \quad \propto \quad B_t A^T \alpha_{t-1}, \quad t = 2:N \tag{11}$$

$$\xi_{t|t}(i,j) \quad = \quad B_t(j)A(i,j)\alpha_{t-1}(i) \tag{12}$$

Then recurse backwards as follows. Base case:

$$s^*_T \sim p(S_T|y_{1:T}) = \alpha_T(S_T) \tag{13}$$

Induction step:

$$s^*_t \quad \sim \quad p(S_t|s^*_{t+1:T}, y_{1:T}) \tag{14}$$

$$\propto \quad p(S_t|s^*_{t+1}, y_{1:t}) \tag{15}$$

We can compute the sampling distribution as follows

$$p(S_t = i|S_{t+1} = j, y_{1:t})\xi_{t|T}(i,j) \quad = \quad \frac{\xi_{t+1|t+1}(i,j)}{\alpha_{t+1}(j)} \tag{16}$$

It will also be useful (see Section 2.2.7) to simultaneously compute the smoothed state estimates in the backwards pass as follows. Base case

$$\gamma_T(j) \quad \propto \quad \alpha_T(j) \tag{17}$$

Induction step

$$\gamma_t(j) \quad = \quad \sum_i \xi_{t|t}(i,j)\frac{\gamma_t(j)}{\alpha_t(j)} \tag{18}$$

2.2.2 Updating O

We just use Bayes rule:

$$p(O_i = 1|y_i, \rho_i, \mu_{0:K}, \sigma_{0:K}, s_i) = \tag{19}$$

$$\frac{\rho_i \mathcal{N}(y_i|\mu_0, \sigma_0)}{\rho_i \mathcal{N}(y_i|\mu_0, \sigma_0) + (1-\rho_i)\mathcal{N}(y_i|\mu_{s_i}, \sigma_{s_i})} \tag{20}$$

For later use we will define

$$\omega_i(o) = p(O_i = o|y_i, \rho_i, \mu_{0:K}, \sigma_{0:K}, s_i) \tag{21}$$

2.2.3 Updating μ and σ^2

We update μ and σ^2 jointly since they are coupled in the posterior (and the prior). Recall the following standard result for conjugate updating of a Normal-inverse-Chi-squared model [8]:

$$p(\mu_s, \sigma^2_s|y_{1:T}) \quad = \quad \mathcal{N}(\mu|m'_s, \frac{\sigma^2_s}{\kappa'_s})\chi^{-2}(\sigma^2_s|\nu'_s, \tau^{2'}_s) \tag{22}$$

$$m'_s \quad = \quad \frac{\kappa_s}{\kappa_s + N_s}m_s + \frac{N_s}{\kappa'_s}\overline{y}_s \tag{23}$$

$$\kappa'_s \quad = \quad \kappa_s + N_s \tag{24}$$

$$\nu'_s \quad = \quad \nu_s + N_s \tag{25}$$

$$\nu'_s\tau^{2'}_s \quad = \quad \nu_s\tau^2_s + N_s\hat{\sigma}^2_s + \frac{\kappa_s N_s}{\kappa'_s}(\overline{y}_s - m_s)^2 \tag{26}$$

where, in our case, we sum over all observations that are in state *s and* which are not outliers:

$$N_s \quad = \quad \sum_{i=1}^N I(S_i = s, O_i = 0) = \sum_{i:S_i=s,O_i=0} 1 \tag{27}$$

$$\overline{y}_s \quad = \quad \frac{1}{N_s}\sum_{i:S_i=s,O_i=0} y_i \tag{28}$$

$$\hat{\sigma}^2_s \quad = \quad \frac{1}{N_s}\sum_{i:S_i=s,O_i=0} (y_i - \overline{y}_s)^2 \tag{29}$$

Since the posterior factorizes

$$\sigma_s^2 | y_{1:T} \sim \chi^{-2}(\nu_s', \tau_s') \quad (30)$$

$$\mu_s | \sigma_s^2, y_{1:T} \sim \mathcal{N}(m_s', \sigma_s^2/\kappa_s') \quad (31)$$

we can sample from the posterior by first sampling σ^2 and then sampling from $\mu|\sigma^2$. To sample from a $\sigma^2 \sim \chi^{-2}(\nu, \tau) = IG(\nu/2, \nu\tau/2)$ distribution, we can use

$$1/\sigma^2 \sim Ga(\nu/2, \nu\tau/2) \quad (32)$$

2.2.4 Updating ρ

Recall that updating beta distributions just means incrementing the pseudocounts. For those locations $i \in \mathcal{P}$ known to be covered by a CNP, we have simply

$$\begin{aligned} p(\rho_i|o_i) &= Beta(a_i', b_i') && (33)\\ a_i' &= a_i + I(o_i = 1) && (34)\\ b_i' &= b_i + I(o_i = 0) && (35) \end{aligned}$$

where we can see that we just update the prior with one data point. For the other locations, we just count the total fraction of outliers:

$$\begin{aligned} p(\rho_0|o_{1:n}) &= Beta(a_0', b_0') && (36)\\ a_0' &= a_0 + \sum_{i=1}^{n} I(o_i = 1, i \notin \mathcal{P}) && (37)\\ b_0' &= b_0 + \sum_{i=1}^{n} I(o_i = 0, i \notin \mathcal{P}) && (38) \end{aligned}$$

2.2.5 Updating A

We update the counts as follows

$$\begin{aligned} p(A_{ij}|s_{1:n}) &= Dir(N_{ij} + \delta_{ij}) && (39)\\ N_{ij} &= \sum_{n=1}^{N-1} I(S_n = i, S_{n+1} = j) && (40) \end{aligned}$$

2.2.6 Updating π

We update the counts as follows

$$\begin{aligned} p(\pi_i|s_1) &= Dir(N_i + \delta_i) && (41)\\ N_i &= I(S_1 = i) && (42) \end{aligned}$$

2.2.7 Rao-Blackwellisation

A simple estimate of the state probabilities, $\hat{\gamma}_i(s) = p(S_i = s|y_{1:n})$, can be computed by simply counting the number of samples in which $S_i = s$. However, the following Rao-Blackwellised estimate

$$\hat{\gamma}_i(s) = \frac{1}{N_{samples}} \sum_{k=burnin}^{niter} \gamma_i^k(s) \quad (43)$$

has much lower variance, since it sheds a layer of Monte Carlos variability by averaging probabilities rather than events simulated with those probabilities. (Here $N_{samples} = niter - burnin$.) We can similarly estimate

$$p(O_i = o|y) = \frac{1}{N_{samples}} \sum_{k=burnin}^{niter} \omega_i^k(o) \quad (44)$$

Note that we currently use $burnin = 0$, $niter = 100$.

2.3 Evaluation methods

We evaluated our algorithm by calculating precision and recall for aberrations (gains and losses grouped together). Given a ground truth labeling and a predicted labeling of the clones (obtained by thresholding the $p(S_i|y)$ probabilities), let ntp be the number of true positives (correctly predicted aberrations), let nt be the number of true aberrations, and let np be the number of predicted aberrations. Recall is defined as $\frac{ntp}{np}$, meaning the proportion of true aberrations detected by the algorithm. Precision is defined as $\frac{ntp}{nt}$, meaning the proportion of predicted aberrations that are true. By varying the threshold on the probabilities, we can vary the tradeoff between precision and recall. To summarize the precision-recall curve in one number, we use the F-measure, which is the geometric mean:

$$F = 2 \times \frac{precision \times recall}{precision + recall} \quad (45)$$

To summarise accuracy results over many samples or chromosomes, we use distributions of F-measures.

We now explain how we modify the above method to handle outliers. We first compute the posterior probability of outlier for each clone, $p(O_i = 1|y)$. We then rank these probabilities and take the top $p_o\%$ of them; finally, we select those whose absolute probability is above a threshold t_o. We then remove all those clones, which are deemed outliers, and compute precision-recall on the remaining locations in the usual way. (Currently we do not assess the reliability of outlier detection, since we do not have reliable ground truth for outlier locations for real data; we are interested in considering outliers in order to help detect aberrations.)

3 Results

To systematically test our approach, we ran three variants of our algorithm on each data set:

- The baseline HMM which clamps the probability of outlier at each location to 0, $p(O_i = 1) = 0.0$. This reduces the model to an HMM with no outlier processing ability, as in [9].
- The robust HMM, which uses $C = 0$ CNPs but updates the global outlier probability $p(\rho_0|y)$ given data from all locations.

- The robust HMM augmented with location specific prior (LSP) knowledge. In particular, we allow all locations $i \in P$ to have their own prior probability of outlier, ρ_i.

We also ran MergeLevels, considered to be the current best method.

3.1 Simulated data with outliers

To check our model works, we used the synthetic data created by Willenbrock and Fridlyand [25], downloaded from `http://www.cbs.dtu.dk/~hanni/aCGH/`. This data is fairly realistic, since it is generated by sampling segments from a large set of primary tumours [25]. To simulate CNPs, we modified this data by adding outliers planted randomly at 10% of the locations in the samples. The positions were sampled from a uniform distribution from 1 to 2000 (the number of clones in each sample). The $\log_2$ ratios for these outliers were sampled from a Gaussian distribution with mean 0 and variance 2. This gave us a data set with ground truth locations for the aberrated clones and for the positions of the outliers.

We chose 10% as the outlier fraction for the following reason. Our internally generated list of CNPs covers nearly 20% of the SMRT clones. However, publicly available CNPs represent approximately 1% of the SMRT clones. Therefore, we chose 10% as a reasonable compromise between these extremes.

In synthetic data, we can control what fraction of the known outlier locations we actually choose to incorporate into our prior, to simulate the effect of partial knowledge. We also add locations to the prior which are not outliers, to simulate the effect of an incorrect prior. In addition to choosing the locations, we can choose the strength of the prior. We set the prior probability to 0.75, which empirically worked better than some lower values we tried, possibly because it increases the sensitivity of the model to single clone outliers. (As we see below, such a strong prior does not hurt performance, even when it is wrong.)

Figure 3 shows a sample from the simulated data set with outliers added. In (A) we see the ground truth labeling, in (B) the result of MergeLevels, and in (C), our robust-LSP method, informed with 50% of the known outlier locations. The vertical dashed lines represent the boundaries of the simulated chromosomes. MergeLevels only detected four of eleven aberrated regions, while the Robust-LSP-HMM detected all regions, although with some false positive results. The LSP model was only given locations for half of the outliers, yet easily detects the remaining ones. We have also tried versions where the LSP model is given incorrect prior locations (i.e., a superset of the known outlier locations). It successfully learns to ignore the prior in this case.

To complement this qualitative assessment, in Figure 4 we present distributions of accuracy on 100 samples for the three variants of our algorithm, including the Robust-LSP-HMM informed by a superset of the positions, half of the positions, and exactly all the positions of known outliers. Distributions are shown as box-and-whisker plots where the line within the box indicates the median of the distribution, the top and bottom edges of the box indicate the third and first quartiles, the ends of the whiskers indicate the 95% confidence intervals of the distribution. The points shown on the plots are outside the 95% confidence intervals.

MergeLevels performs considerably worse than all the HMMs: its F-measure was 0.37±0.26 over 100 samples. The Base-HMM had a F-measure of 0.56±0.17, indicating that by using an HMM framework, significant improvement is attained over MergeLevels. Further improvement was attained by adding outlier detection. The Robust-HMM had a F-measure of 0.68±0.14. Finally the three versions of the Robust-LSP-HMM performed increasingly better when informed by a superset of the positions (F-measure 0.69±0.14), half of the positions (F-measure 0.73±0.15) and exactly all the positions (F-measure 0.77±0.15) of the known outliers. This shows that informative prior knowledge can help, but incorrect prior knowledge will not hurt performance (as long as the prior is not too strong, contradictory data will always overwhelm it).

3.2 H526 lung cancer cell line data

To illustrate the performance of our method on real data, we used a set of 12 replicates from the well studied H526 lung cancer cell line and evaluated performance on four chromosomes (1, 3, 4, 17) known to exhibit aberrations [13]. The data was generated using the Sub-Megabase Resolution Tiling (SMRT) arrays [13] using a set of approximately 27,000 clones that cover the human genome. The $\log_2$ ratios were normalised according to the stepwise method described in Khojasteh *et al* [15]. We produced a single 'sample' from the 12 replicates by taking the median $\log_2$ ratio for each clone after removing between-array systematic variations at each position. This was then manually labeled. This resulted in a relatively clean data set in terms of noise, but it allowed us to test our model on high resolution real data, likely to contain CNPs.

We used a publicly available list of CNPs available from `http://projects.tcag.ca/variation/`, first described in [12], and an internally generated list of CNPs (Wong *et al*, unpublished) detected using SMRT arrays on a population of 105 normal individuals to set the location specific prior probability of an outlier. We report results using our internally generated CNPs.

Figure 5 shows chromosome 3 of H526 labelled by an expert (A), by MergeLevels (B) and by Robust-LSP-HMM (C). We show results from chromosome 3 as it shows the biggest difference between MergeLevel output and the Robust-LSP-HMM output; the differences on other chromosomes are less dramatic.

MergeLevels mis-classifies a large neutral region on the p-arm as a loss and misses several small aberrations on

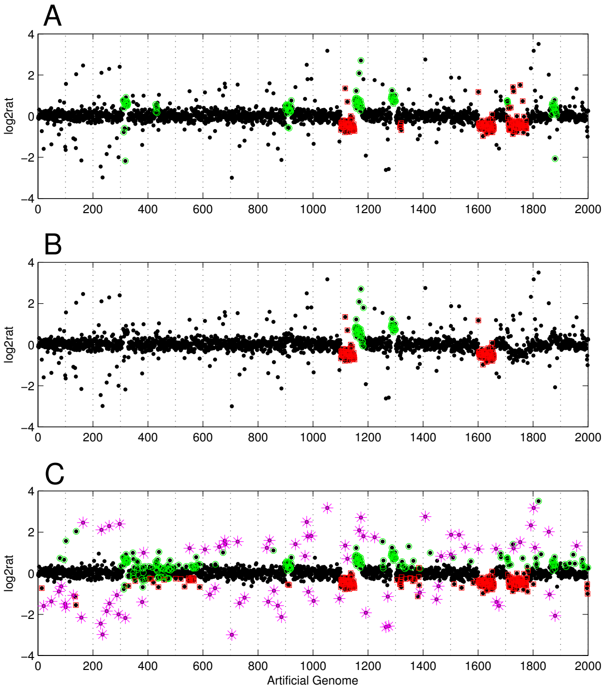

Figure 3: Array CGH profile for a sample from the simulated data set of Willenbrock and Fridlyand [25] augmented with outliers. The color scheme is the same as explained in Figure 5. Panel **A** shows the ground truth labeling of the sample. Vertical dashed lines represent the boundaries of the chromosomes. Each chromosome was analysed separately. The outlying points were inserted randomly at 10% of the locations according to a Gaussian distribution with μ=0 and σ=2. There are 11 aberrated regions in this sample. Panel **B** shows the output of DNACopy+MergeLevels. This only detected 4 of the aberrated regions. Panel **C** showing the output of the Robust-LSP-HMM informed by half of the locations of outliers. All 11 aberrated regions are detected by our algorithm, although there are numerous false positive predictions. We set p_o=10% and t_o=0.1 when detecting outliers.

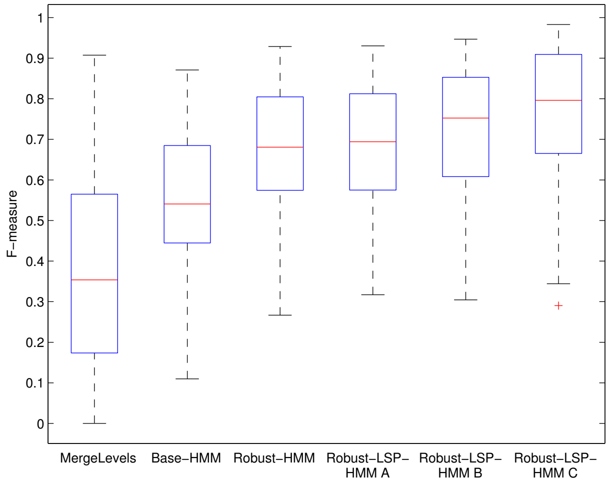

Figure 4: F-measures for 100 samples of Willenbrock and Fridlyand's simulated data augmented with outliers. From left to right: MergeLevels had an F-measure of 0.37±0.26. The Base-HMM had better accuracy (F-measure of 0.56±0.17). Further improvement was gained using the Robust-HMM (F-measure of 0.68±0.14). As expected, informing the Robust-LSP-HMM with the locations of the outliers resulted in the best performance. Robust-LSP-HMM **A** (F-measure=0.69±0.14) was informed with a superset of the outlier locations, Robust-LSP-HMM **B** (F-measure 0.73±0.15) was informed with half of the locations, and Robust-LSP-HMM **C** (F-measure 0.77±0.15) was given all and only the outlier locations.

the q-arm, but is otherwise correct. Also note the obvious positive outliers near the center of the chromosome that MergeLevels mis-classified as losses. The F-measure was 0.73 for MergeLevels.

Our algorithm performed better than MergeLevels on chromosome 3 (F-measure = 0.94) and nearly matched the ground truth labeling, with the exception of missing very small aberrations. The algorithm was given the complete list of CNPs described in Section 3.2 that covered approximately 20% of the clones. We used $p_o = 1\%$ and t_o=0.1 to determine outliers. Other parameters used for this data set are listed in Table 1.

Figure 6 shows the quantitative evaluation of over chromosomes 1, 3, 4, and 17 of H526 for MergeLevels, the Base-HMM with no outlier detection, the Robust-HMM and the CNP-informed Robust-LSP-HMM. The HMMs performed marginally better than MergeLevels: The F-measures for the HMMs are 0.96±0.02, while for MergeLevels it is 0.91±0.12. In view of the big differences on the synthetic dataset, we were somewhat surprised that MergeLevels did nearly as well as the HMMs on average. (The low minimum performance of MergeLevels is mainly attributed to the misclassification of the large region on the p-arm of chromosome 3 mentioned earlier.) We believe this is because our synthetic data is actually harder than this real data set. However, recall that this “real” data set is actually the result of averaging 12 replicates, and hence is less noisy than one would typically expect. (Unfortunately, we only had ground truth labels for the averaged data set.)

Another thing to note from Figure 6 is that all the HMM variants do basically the same. This is a strong result, considering that the list of CNPs covers about 20% of the clones. Once again, this indicates that the algorithm allows the data to overwhelm the prior at CNP locations that are not exhibited in the sample. In addition, note that although the performance of the regular HMM and the robust methods were equivalent, the robust methods provide extra information over the regular HMM, since they flag outliers.

4 Discussion

We have presented a new model for classifying aberrated clones in aCGH data, which is robust to outliers and is able to leverage prior knowledge about CNP locations. In simulated data, this model works better than a standard HMM and a state of the art method, DNAcopy+MergeLevels.

However, on real H526 data, all methods work about the same, on average. We believe one reason for this may be the evaluation metric. We evaluated performance at the level of precision/ recall of individual clones (as is standard). Thus methods that get most of the large aberrations correct will score well, even if they miss smaller aberrations (since the small ones constitute a small fraction of the test set). However, it may be more clinically relevant to measure precision/ recall of aberrated regions rather than aberrated clones. This will then treat all regions equally, regardless of size, and will reward algorithms that can detect narrow aberrations. Narrow aberrations are by their nature harder to detect. We believe our method, which can incorporate prior knowledge, stands a better chance of detecting these. (Of course, many of these small aberrations may be called outliers, so we may need to merge the outlier and gain labels.)

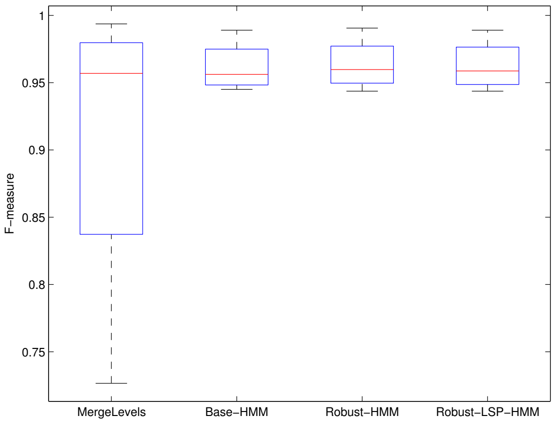

Figure 6: F-measures over chromosomes 1,3,4,17 for MergeLevels, the Base-HMM, the Robust-HMM and the Robust-LSP-HMM with CNP location prior. The F-measures is lower (mean 0.91±0.12) for MergeLevels than the HMM variants, whose F-measures were all 0.96±0.02. (Note that the horizontal line inside a box plot denotes the median, not the mean.)

In the future, we plan to apply the method to samples extracted from a cohort of lymphoma patients. The aCGH profiles for these patients have been manually classified and numerous clinically relevant aberrations have been identified. We will evaluate applicability of our method in a clinical setting using this data set.

We are also developing new models to identify locations of recurrent aberrations across samples, and to use other forms of prior knowledge, such as the locations of fragile sites. Combined with CNP information, we anticipate that such models will be extremely useful in profiling sub-types of cancer with aCGH.

Acknowledgement

We thank Kevin Leyton-Brown from the UBC Computer Science department for providing computing resources. This work is supported by a grant from Genome Canada.

References

[1] P. Broet and S. Richardson. Detection of gene copy number changes in CGH microarrays using a spatially correlated mixture model. *Bioinformatics*, 2006.

[2] P G Buckley, K K Mantripragada, A Piotrowski, T Diaz de Ståhl, and J P Dumanski. Copy-number polymorphisms: mining the tip of an iceberg. *Trends Genet*, 21(6):315–317, Jun 2005.

[3] Bradley P. Carlin and Thomas A. Louis. *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman and Hall, 1996.

[4] L. Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag, 1986.

[5] PH. Eilers and RX. de Menezes. Quantile smoothing of array CGH data. *Bioinformatics*, 21(7):1146–1153, Apr 2005.

[6] D A Engler, G Mohapatra, D N Louis, and R A Betensky. A pseudolikelihood approach for simultaneous analysis of array comparative genomic hybridizations (acgh). *Biostatistics*, Jan 2006.

[7] J. Fridlyand, A. Snijders, D. Pinkel, D. Albertson, and A. Jain. Hidden Markov Models Approach to the Analysis of Array CGH data. *J. Multivariate Analysis*, 40:132–153, 2004.

[8] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian data analysis*. Chapman and Hall, 2004. 2nd edition.

[9] S. Guha, Y. Li, and D. Neuberg. Bayesian Hidden Markov Modeling of Array CGH Data. Technical report, Harvard School of Public Health, 2006.

[10] L. Hsu, SG. Self, D. Grove, T. Randolph, K. Wang, JJ. Delrow, L. Loo, and P. Porter. Denoising array-based comparative genomic hybridization data using wavelets. *Biostatistics*, 6(2):211–226, Apr 2005.

[11] P. Hupé, N. Stransky, JP. Thiery, F. Radvanyi, and E. Barillot. Analysis of array CGH data: from signal ratio to gain and loss of DNA regions. *Bioinformatics*, 20(18):3413–3422, Dec 2004.

[12] A J Iafrate, L Feuk, M N Rivera, M L Listewnik, P K Donahoe, Y Qi, S W Scherer, and C Lee. Detection of large-scale variation in the human genome. *Nat Genet*, 36(9):949–951, Sep 2004.

[13] AS. Ishkanian, CA. Malloff, SK. Watson, RJ. DeLeeuw, B. Chi, BP. Coe, A. Snijders, DG. Albertson, D. Pinkel, MA. Marra, V. Ling, C. MacAulay, and WL. Lam. A tiling resolution DNA microarray with complete coverage of the human genome. *Nat Genet*, 36(3):299–303, Mar 2004.

[14] K Jong, E Marchiori, G Meijer, A V Vaart, and B Ylstra. Breakpoint identification and smoothing of array comparative genomic hybridization data. *Bioinformatics*, 20(18):3636–3637, Dec 2004.

[15] M Khojasteh, W L Lam, R K Ward, and C MacAulay. A stepwise framework for the normalization of array cgh data. *BMC Bioinformatics*, 6:274–274, Nov 2005.

[16] W. Lai, M. Johnson, R. Kucherlapati, and P. Park. Comparative analsysis of algorithms for identifying amplifications and deletions in array CGH data. *Bioinformatics*, 2005.

[17] AB. Olshen, ES. Venkatraman, R. Lucito, and M. Wigler. Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, 5(4):557–572, Oct 2004.

[18] F Picard, S Robin, M Lavielle, C Vaisse, and JJ Daudin. A statistical approach for array CGH data analysis. *BMC Bioinformatics*, 6(1):27–27, Feb 2005.

[19] D. Pinkel and D. Albertson. Array comparative genomic hybridization and its application in cancer. *Nature Genetics Supplement*, 37:S11–S17, 2005.

[20] L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–286, 1989.

[21] C. Robert. Simulation of truncated normal distributions. *Statistics and computing*, 5:121–125, 1995.

[22] C. Rouveirol, N. Stransky, Ph. Hupe, Ph. La Rosa, E. Viara, E. Barillot, and F. Radvanyi. Computation of recurrent minimal genomic alterations from aCGH data. *Bioinformatics*, 22(7), 2006.

[23] S Scott. Bayesian methods for hidden markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, 2002.

[24] J Sebat, B Lakshmi, J Troge, J Alexander, J Young, P Lundin, S Månér, H Massa, M Walker, M Chi, N Navin, R Lucito, J Healy, J Hicks, K Ye, A Reiner, T C Gilliam, B Trask, N Patterson, A Zetterberg, and M Wigler. Large-scale copy number polymorphism in the human genome. *Science*, 305(5683):525–528, Jul 2004.

[25] H. Willenbrock and J. Fridlyand. A Comparison Study: Applying Segmentation to Array CGH Data for Downstream Analysis. *Bioinformatics*, 2005.

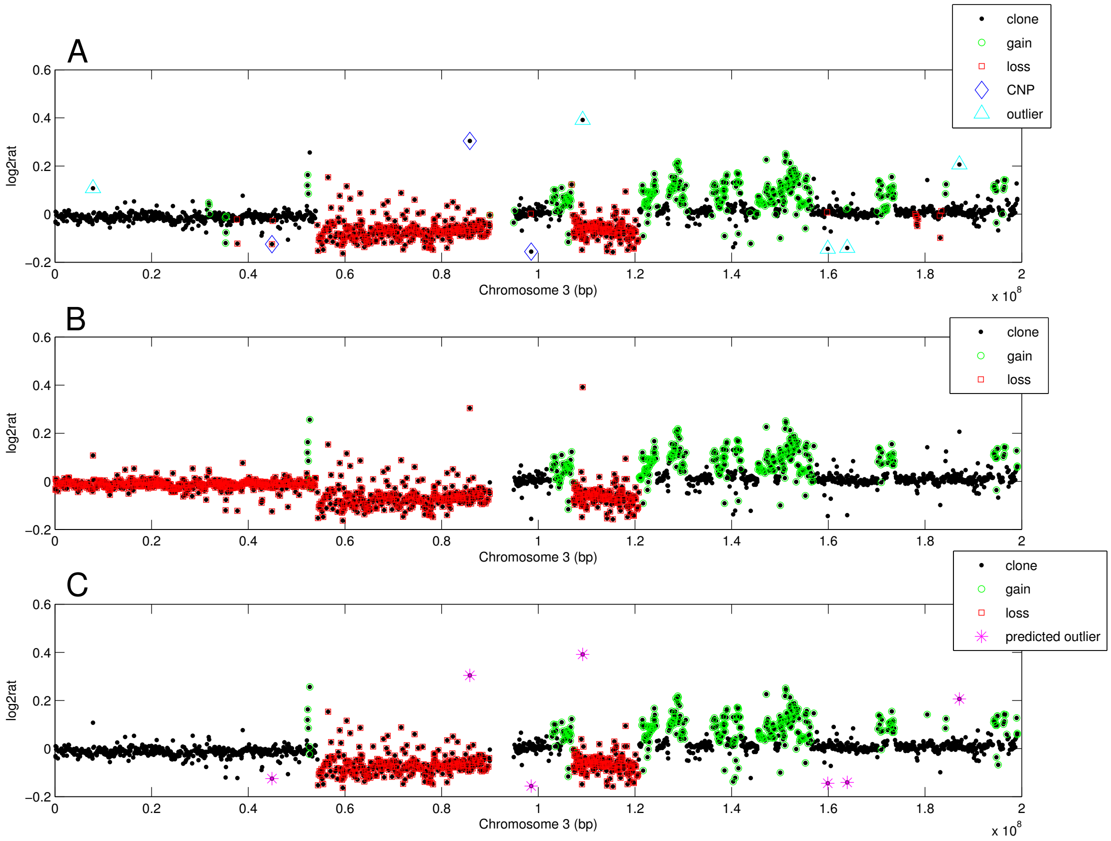

Figure 5: Array CGH profile for chromosome 3 of the lung cancer cell line H526. Panel **A** shows the $\log_2$ ratios plotted against the position of each clone on the chromosome. The red squares indicate clones labeled as losses by an expert cytogenetecist. The green circles similarly indicate clones that are gains. Clones marked with dark blue diamonds indicate a known CNP (for clarity, not all CNPs are shown). Clones marked with light blue triangles indicate non-CNP outliers identified by eye. These may represent single clone aberrations, measurement errors or previously unknown CNPs. Panel **B** shows the predicted gains and losses output by MergeLevels (F-measure 0.73). Panel **C** shows the output of the Robust-LSP-HMM (F-measure 0.94). Predicted outliers are shown as pink stars. Note that the algorithm finds both CNP and non-CNP outliers (marked as such in Panel **A**), while correctly identifying nearly all aberrated clones.