Stony Brook University

A Comparison of Hidden Markov Model Based Programs for Detection of Copy Number Variation in Array Comparative Genomic Hybridization Data

A Dissertation Presented

by

Andrea Roberson

to

The Graduate School

in Partial Fulfilment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

May 2010

Stony Brook University

The Graduate School

Andrea Roberson

We, the dissertation committee for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation.

Stephen J. Finch - Dissertation Advisor
Professor of Statistics
Department of Applied Mathematics and Statistics

Nancy R. Mendell - Chairman of Defense
Professor of Statistics
Department of Applied Mathematics and Statistics

Wei Zhu
Professor of Statistics
Department of Applied Mathematics and Statistics

Derek Gordon
Department of Genetics
Rutgers University

This dissertation is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

A Comparison of Hidden Markov Model Based Programs for Detection of Copy Number Variation in Array Comparative Genomic Hybridization Data

by

Andrea Roberson

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

2010

Array comparative genomic hybridization (aCGH) can detect copy number variation (CNV) across the genome. Five current Hidden Markov Model (HMM) software systems for estimating copy number variation with aCGH data were compared. These comparisons were in terms of their effectiveness for identifying CNVs in simulated data based on the ratio of signal intensities. There was significant variability in the error rates. The system that adjusted for outliers in the model, the Robust Hidden Markov Model (HMM-R), appeared to have the best performance. The emission density function of the HMM is a mixture of two normal densities, in which one component represents usable aCGH data and the other represents outliers. HMM-R correctly classified 99.8% of normal states, 84.5% of CNV gains, and 90.2% of CNV losses.

That is, error rates with regard to gains and losses were appreciable even with the best software. The HMM-R method demonstrated higher sensitivity and lower false discovery rates than the commonly used procedure. While the accuracy rates of HMM software has improved, there is substantial room for further improvement.

To York, Sandra, Devin, Janie, and Janel

with all my love

Table of Contents

List of Figures

List of Tables

Chapter 1

Background and Introduction

1.1 Copy Number Variation

Studying human genetic variation allows us to understand the complex mechanisms by which DNA sequences impact disease. Variations among the genomes of different individuals arise by mutation, which is a structural alteration of the DNA (Freeman et al., 2006). Large segments of DNA, ranging in size from thousands to millions of DNA bases, can vary in copy-number. Feuk et al. (2006) defines Copy Number Variations (CNVs) as duplications or deletions of a segment of DNA sequence compared to a reference genome.

Geneticists have long been aware of large-scale deletions and duplications (e.g. Ford et al. 1959, Summitt 1964). One of the first observations was that children with Down's syndrome have an additional copy of chromosome 21 (Jacobs et al., 1959). Prior to 2004, these major rearrangements were considered to be rare events. However, recent discoveries have revealed that genomic imbalances that do not result in genetic disorders actually occur much more frequently. Redon et al. (2006) constructed a CNV map of the genome, which encompassed 270 DNA samples, and identified over 1,400 copy number variable regions, about 12% of the human genome. Most CNVs are benign variants that seem not to cause disease directly. However, there are several instances where CNVs that affect critical developmental genes do cause disease. Gonzalez et al. (2005) recently

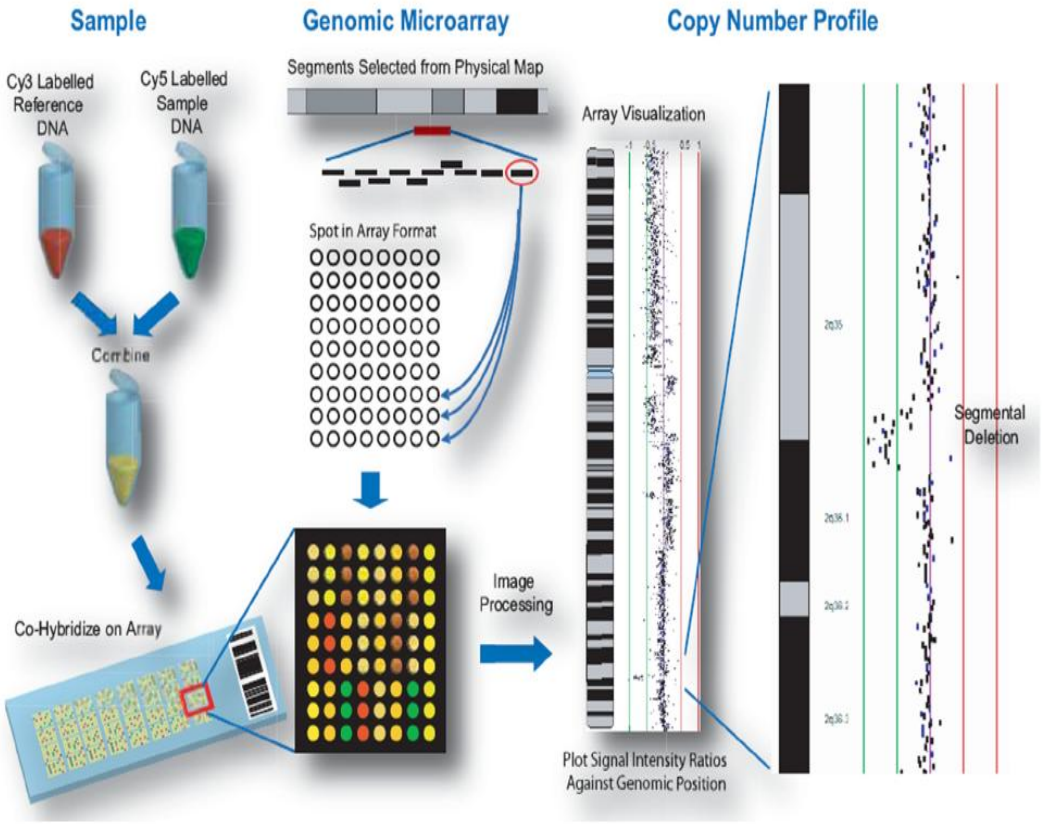
showed that the number of copies of the CCL3L1 gene influences susceptibility to HIV / AIDS. Sebat et al. (2007) reported that CNVs appear to create a greater risk for autism.

CNVs have also been associated with cancer development and progression (Shlien and Malkin, 2010). Oncogenes cause the cells to divide at a rapid rate, resulting in tumors. Copy number gains appear to lead to overexpression of oncogenes. Tumor suppressor genes tend to hold the cells back, inhibiting mitosis when there are cell defects. Copy number losses lead to underexpression of tumor suppressor regions. The HER-2 (human epithelial receptor 2) oncogene resides on chromosome 17q, and is involved in cell growth and development. Amplification of the HER-2 gene can be detected in 20-30% of invasive breast cancers (Ordas et al., 2007). This amplification is associated with aggressive tumors with a poor prognosis. The discovery of the HER-2 gene amplification has led to HER-2 targeted treatments and therapeutic applications. This CNV defines a subgroup of high-risk breast cancer patients who benefit from individually tailored chemotherapy drugs such as Adriamycin (Tanner et al., 2006).

1.2 Method for detecting Copy Number Variation

Array comparative genomic hybridization (aCGH) is an array-based high-resolution method to detect copy number variation. The technology seeks to detect and map chromosomal aberrations, on a genomic scale, in a single experiment (Picard et al., 2005). For whole-genome aCGH, genomic DNA isolated from a test and a reference sample are fluorescently labeled with two different colored dyes (Figure 1).

Figure 1 To perform array-CGH, DNA samples from the reference and control are labeled with different colors of fluorescent dye (green for sample DNA, red for reference DNA), then mixed together. Each mix is matched (hybridized) to a separate slide with DNA clones. A laser scanner reads both fluorescent signals, and a log ratio of intensities is produced for each probe on the array. The graph below shows the plot of log ratios as a function of location on the corresponding probe.



Source: Shah et al. 2006

The DNA is allowed to hybridize to a microarray of whole-genome clones attached to a polymer-coated glass slide. After the slides have been incubated and washed, a laser scanner reads them to determine the relative strength of the two fluorescent colors at each of the DNA spots on the array. Image analysis then results in test and reference intensities for all array elements. The intensity of an array element is linearly proportional to the abundance of the corresponding DNA sequence in the sample. The

$\log_2$ ratio of the test and reference intensities reflect the relative copy number in the test sample compared to that in the reference sample.

1.3 Pre-processing

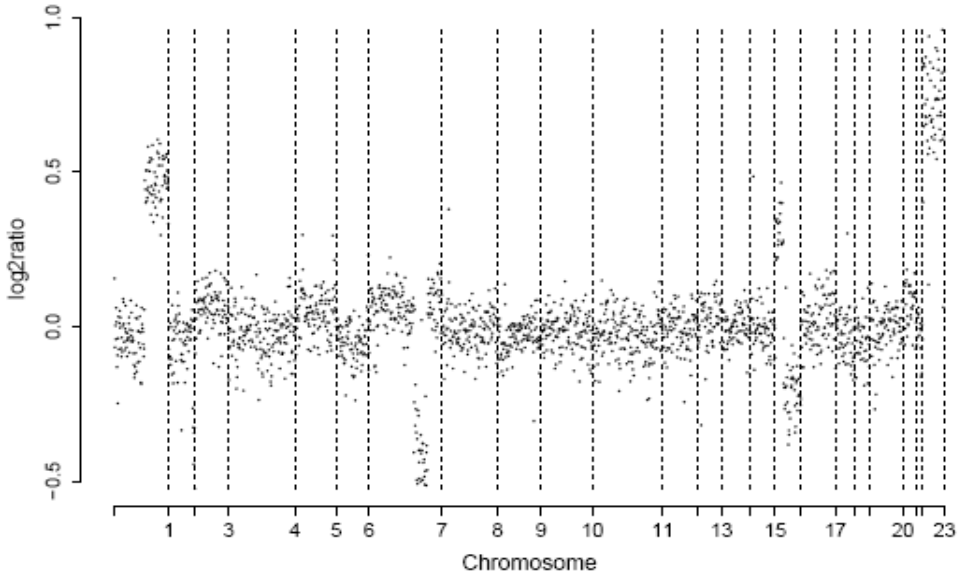
For aCGH, the $\log_2$ ratios undergo three pre-processing steps before arriving at the actual copy number (Van Wieringen et al., 2008). The three steps of pre-processing include normalization, segmentation, and calling. With all microarray-based techniques, the fluorescence intensity ratios first have to be normalized to correct for non-biological sources of error such as intensity fluctuations, background noise and fabrication artifacts (Brown et al., 2001). The next step, segmentation, is the process by which the boundaries for copy number alterations are determined. Chromosomes are routinely divided into segments of constant copy number. In general, the formulation of a model-based method presumes a sequence of piece-wise constant segments as a function of various parameters such as the number of breakpoints, their locations and the mean/variance of the distributions for each segment (Lai et al., 2005). Estimating the true sequence of underlying copy number ratios that generated the observed sequence of fluorescence ratios is the final stage of the process, or the calling step.

Array-based CGH data consists of $\log_2$ transformed fluorescence intensity ratios, which are linearly proportional to the copy numbers. In the absence of normalization or measurement errors, the normal clones would correspond to a $\log_2$ ratio of zero, because the normal and tumor DNA fragments both have two copies (Guha et al., 2008). Genomic regions of copy number gains should have ratios greater than or equal to 3/2

(i.e, $\log_2(3/2) \geq 1.58$); and regions of copy number loss should have ratios equal to 1/2 (i.e, $\log_2(1/2) = -1$). Logarithms of the ratios are commonly used because the ratios are dependent on absolute magnitude and are often highly skewed. Logged intensities often provide a better sense of the true variation (Engler et al., 2006).

Figure 2 displays the normalized $\log_2$ ratios of a breast cancer specimen analyzed by Snijders et al. (2001). The data highlight the necessity of applying statistical techniques to arrayCGH data. After normalization there is considerable shrinkage of the $\log_2$ ratios towards zero. The plot illustrates the deviation of these $\log_2$ ratios from the theoretical values.

Figure 2 Normalized copy number ratios of a comparison of DNA from tumor cell strain S0034 with normal DNA. The clones are ordered by position in the genome. The vertical bars indicate borders between chromosomes.



Source: Snijders et al. 2001

Most calling algorithms in the literature are unable to elicit a segment's actual copy number. They do, however, detect deviations from the normal copy number and

classify each segment as either 'normal', 'loss', 'gain', or 'amplification': 'normal' if there are two copies of the chromosomal segment present, 'loss' if at least one copy is lost, 'gain' if at least one additional copy is present, and amplification if there are a high level of copies present (Van Wieringen et al., 2008).

1.4 Goals of this project

Fridlyand et al. (2004) were the first to consider Hidden Markov Models (HMM) for calling aberrant CNVs. A number of HMMs have since been developed for the purpose of copy number analysis (Marioni et al. 2006, Shah et al. 2006, Stjernqvist et al. 2007). In recent years, two survey papers by Willenbrock and Fridlyand (2005) and Lai et al. (2005) have been published comparing various methods for finding copy number segments. Lai et al. (2005) reported that the most effective method for finding copy number segments is Circular Binary Segmentation (CBS) (Olshen et al., 2004). Lai et al. (2005) found that the HMM results were sub-optimal with high false discovery rates (~40-60%), and lower sensitivity (~50-80%). Willenbrock and Fridlyand (2005) reported parallel findings. However, HMMs have been widely implemented, offering benefits such as fast computational speeds, which is the primary reason they are the most widely implemented methods today.

In the last five years, Bayesian methods have been incorporated in HMM analysis (e.g., Engler et al., 2006, Shah et al., 2006, Rueda and Diaz-Uriarte, 2007, Guha et al., 2008). Markov Chain Monte Carlo (MCMC) strategies are used to infer model parameters (Scott, 2002), and the optimal state sequence is determined from the estimated Bayesian posterior distribution. Copy number gains and losses are identified from these distributions.

Shah et al. (2006) demonstrated higher accuracies for their Bayesian HMM that were comparable to CBS (Olshen et al., 2004). The purpose of this study is to compare the performance of five current HMM programs: BioHMM (Marioni et al., 2006), HMM-R (Shah et al., 2006), Bayesian HMM (Guha et al., 2008), CGHclassify (Engler et al., 2006), and RJaCGH (Rueda and Diaz-Uriarte, 2007) using simulated data.

Chapter 2

Hidden Markov Models

2.1 Overview of HMM

It is useful to visualize a HMM generating a sequence. The model moves through a series of states and produces output either when it has reached a particular state or when it is moving from state to state (Eddy, 2004). The state path is a Markov chain, meaning that the next state depends only on the current state. The HMM approach seems promising since its model incorporates DNA copy number transitions.

2.2 A HMM for copy number assignments

Clearly defined parameters must be specified to call copy numbers from a sequence of continuous $\log_2$ ratios. Using a HMM, the sequence of clones is traversed in one direction only, according to chromosomal position, moving between the hidden copy number states in the model (Andersson et al., 2008). For the case when the random variable has a finite state space, we can specify a HMM by the following (Rabiner, 1989):

(1) A set of *n* distinct states $s_1, s_2 \ldots, s_n$. These states model the CNV. There are discrete time steps, *t*= 0, *t*= 1, …. such that at timestep *t* the system is in exactly one of the available states, called q_t, where $q_t \in \{s_1, s_2 \ldots, s_n\}$. The "timesteps" model the progression through the genome.

(2) A distribution of initial states $\pi = \{\pi_i\}$, where $\pi_i = \mathrm{P}(q_0 = s_i)$ is the probability of starting in copy number state *i*.

(3) A set of allowed transitions between states. There is a probability that the transition from state *i* to state *j* is taken. These are the transition probabilities. This probability is usually represented as $A = \{a_{ij}\}$ where

$$P(q_{t+1} = s_j \mid q_t = s_i) = a_{ij} \quad (i, j \le n)$$

(4) In a HMM the states are not observable. When a state is visited, an observation *x* is recorded. For continuous observations, the emission probability density function (pdf) in state *j*, $B = \{b_j(x)\}$, is defined as, $b_j(x) = P(x \mid q_t = s_j)$. The emission pdfs characterize the likelihood of a certain observation, if the model is in state j. Here it characterizes the likelihood of observing a specific intensity ratio. The emission pdf of each state is often assumed to be normally distributed, i.e. $b_j(x) \sim N(\mu_j, \sigma_j^2)$.

Thus, a HMM with a finite number of states can be characterized in terms of three sets of parameters: (i) the initial state probabilities, π; (ii) the transition probability matrix, *A*; (iii) the collection of emission pdfs defined within each state, *B*. Then $\lambda = (A, B, \pi)$ is taken as the parameter set of a HMM.

For most problems, there are so many possible state sequences that one could not practically enumerate them. There are several possible ways of finding the optimal state sequence associated with the given observation sequence. The Viterbi algorithm (1967) is a dynamic programming algorithm, guaranteed to find the most probable state path given a sequence and a HMM. The Viterbi algorithm is used to find the state sequence with the highest probability. The posterior probability of a set of model parameters (λ) given the observations (*x*) is defined to be (Gilks et al., 1996):

$$p(\lambda \mid x) = \frac{L(x \mid \lambda)\, p(\lambda)}{p(x)},$$

where $p(\lambda \mid x)$ is the posterior probability density, $L(x \mid \lambda)$ is the likelihood function, $p(\lambda)$ is the prior probability density, and $p(x)$ is the marginal probability density. The Viterbi algorithm is used to find a maximum posterior probability state sequence; that is, a sequence $Q = (q_1, ..., q_T)$ maximizing $P(Q \mid x, \lambda)$.

2.3 Bayesian Hidden Markov Models

2.3.1 Parameter Estimation

The most important and difficult problem in HMMs is to find the model parameters $\lambda = (A, B, \pi)$ from the data. Here we want to adjust the model parameters to fit the observations best. There is no known way to solve analytically for the parameter set that maximizes the probability of the observation sequence in a closed form (Rabiner and Juang, 1993). The standard approach is to use the Baum-Welch method (Baum et al., 1970) to choose $\lambda = (A, B, \pi)$ such that its likelihood is locally maximized.

2.3.2 Markov chain Monte Carlo Methods

Bayesian Markov chain Monte Carlo (MCMC) sampling strategies can be used to simulate HMM parameters from their posterior distribution given observed data (Scott, 2002). MCMC methods attempt to simulate direct draws from some complex distribution of interest. One starts with some initial parameter value. Then each new parameter value is generated from a probability density that depends on the previous value. The resulting sequence of parameter values forms a Markov chain. The

Metropolis-Hastings algorithm (Hastings, 1970) and the Gibbs sampler (Gelfand and Smith, 1990) are the two major methods in MCMC.

2.3.3 Stochastic Forward-Backward algorithm

The major component of current Bayesian MCMC approaches to HMMs is the simulation of the states from the marginal posterior probability distribution of the state sequences, $P(S \mid Y, \lambda)$ (Chib, 1996). Earlier methods applied the Gibbs sampler, where the individual hidden states are individually sampled in separate Gibbs steps. Sampling a single state at a time introduces many more elements into the Gibbs Markov chain, a less efficient procedure. The preferred alternative to a Gibbs sampling procedure is to use a stochastic version of the forward-backward algorithm (Scott, 2002). The forward-backward algorithm is an efficient method to sample from the posterior distribution of a HMM. The approach samples the whole state sequence, as a single component block, from its posterior distribution directly.

The Stochastic Forward-Backward algorithm (Scott, 2002) consists of the following steps:

(1) Given the observed $\log_2$ ratios $Y = \{y_t\}_{t=1}^{T}$ the forward step calculates the likelihood of the $\log_2$ ratios, $P(Y \mid \lambda)$

(2) The backward step samples the final state, S_N from $P(S_N = i \mid y_1, ...y_N)$

(3) The remaining states $S_{N-1}, ..., S_1$ are sampled recursively from

$P(S_t = i \mid S_{t+1}, ..., S_N, y_1, ...y_N)$

I have developed a simple example to explain estimating the hidden state sequence with the Forward-Backward algorithm of the Bayesian HMM (Guha et al., 2008). The parameter set of the HMM is given in Figure 3. For this system, the state is the chromosomal copy number- loss state, normal state, gain state, or amplification state. The probabilities $P(q_{t+1} = s_j \mid q_t = s_i) = a_{ij}$ $(i, j \leq 4)$ of the next state based on the current state are summarized in the transition matrix *A*. For example, the probability of a gain state followed by a loss state is .19, and the probability that a loss state is followed by a gain state is .56. The transition matrix is row stochastic, meaning that each element is a probability and the elements of each row sum to 1.

Although we can not observe the copy number state, we can observe the $\log_2$ ratios. Conditional on the copy number states, the $\log_2$ ratios are assumed to be distributed as $N(\mu_{s_k}, \sigma_{s_k})$, where $k = 1,...4$. The parameters for the emission pdfs associated with each state are

$\{s_1 \sim N(-.396,.408), s_2 \sim N(.052,.367), s_3 \sim N(.326,.393), s_4 \sim N(1.546,.762)\}$. We can calculate the likelihood of the first $\log_2$ ratio -.3132, given the HMM is in the loss state s_1,

$$f(y;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(y-\mu)^2}{2\sigma^2}}$$

$$f(-.3132;-.396,.408) = \frac{1}{\sqrt{2\pi}(.408)} e^{\frac{-[-.3132-(-.396)]^2}{2(.408^2)}} = .96$$

The likelihood of the $\log_2$ ratio -.3132, given the HMM is in the loss state is .96. This value corresponds to the first element of the Emission matrix *B*.

Figure 3 An example I have developed of a continuous HMM with four states. The copy number state s_k takes values from the set {1,2,3,4}. The value s_k =1 represents a copy number loss; s_k =2 represents the normal state; s_k =3 represents a single copy gain; s_k =4 represents an amplification. The initial parameters I computed from the Bayesian HMM for this example are given here.

Π=[.16 .28 .27 .29]

Transition Matrix- A				
	1	2	3	4
1	.21	.06	.56	.17
2	.002	.29	.11	.60
3	.19	.54	.13	.14
4	.25	.17	.38	.20

Emission - B				
1	.96	.73	.56	.20
2	.66	1.01	1.09	.82
3	.27	.59	.77	1.01
4	.03	.05	.07	.15
	Y_1	Y_2	Y_3	Y_4

Y={-.3132, -.0855, .0346, .3261}

Given the model $\lambda = (A, B, \pi)$ and the sequence of observations $Y = \{-.3132, -.0855, .0346, .3261\}$, the algorithm first calculates the $P(Y \mid \lambda)$. The likelihood of an observation sequence $Y = \{-.3132, -.0855, .0346, .3261\}$ with respect to a HMM with parameters λ expands as

$$\begin{aligned} P(Y \mid \lambda) &= P(\{-.3132, -.0855, .0346, .3261\} \mid \lambda) \\ &= \sum_{\text{all Q}} P(\{-.3132, -.0855, .0346, .3261\}, Q \mid \lambda) \\ &= \sum_{\text{all Q}} P(\{-.3132, -.0855, .0346, .3261\} \mid Q, \lambda) P(Q \mid \lambda) \end{aligned}$$

The probability that $Y = \{-.3132, -.0855, .0346, .3261\}$ was generated by a given model λ is the sum of the joint likelihood of the observation sequence Y and path Q, over all possible state paths Q allowed by the model. The probability that Y and Q occur simultaneously, decomposes into the product of two quantities. The first quantity $P(\{-.3132, -.0855, .0346, .3261\} \mid Q, \lambda)$, is the likelihood of the observation sequence

given the state sequence. It is the product of the emission densities computed along the considered path;

$$P(Y \mid Q, \lambda) = P(\{y_1, y_2, y_3, y_4\} \mid \{q_1, q_2, q_3, q_4\}, \lambda)$$
$$= \prod_{t=1}^{4} P(y_t \mid q_t, \lambda) = b_{q_1}(y_1) \cdot b_{q_2}(y_2) \cdot b_{q_3}(y_3) \cdot b_{q_4}(y_4)$$

The second quantity $P(Q \mid \lambda)$, is the probability of a state sequence $Q = \{q_1, q_2, q_3, q_4\}$ coming from a HMM with parameters λ. This probability corresponds to the product of the transition probabilities from one state to the following.

$$P(Q \mid \lambda) = \pi_{q_1} \cdot a_{q_1, q_2} \cdot a_{q_2, q_3} \cdot a_{q_3, q_4}$$

The joint likelihood of the sequence $Y = \{-.3132, -.0855, .0346, .3261\}$ and the path $Q = \{loss, loss, gain, gain\}$ can be computed from

$$P(Y, Q \mid \lambda) = P(\{-.3132, -.0855, .0346, .3261\}, \{loss, loss, gain, gain\} \mid \lambda)$$
$$= P(\{-.3132, -.0855, .0346, .3261\} \mid \{loss, loss, gain, gain\}, \lambda) P(\{loss, loss, gain, gain\} \mid \lambda)$$
$$= \pi_{loss} \cdot b_{loss}(-.3132) \cdot a_{loss,loss} \cdot b_{loss}(-.0855) \cdot a_{loss,gain} \cdot b_{gain}(.0346) \cdot a_{gain,gain} \cdot b_{gain}(.3261)$$
$$= (.16)(.96)(.21)(.73)(.56)(.77)(.13)(1.01) = .0013$$

The interpretation of the computation in the above equation is the following. Initially (at time *t=1*) we are in the loss state with probability π_{loss}, and generate the $\log_2$ ratio -*.3132* (in this state) with the likelihood $b_{loss}(-.3132)$. At the next time-step (*t*=2) we make a transition to another loss state with probability $a_{loss,loss}$, and generate the $\log_2$ ratio -*.0855* with the likelihood $b_{loss}(-.0855)$. This process continues sequentially until we make the

last transition (at time *t=4*) from the gain state to another gain state with probability $a_{gain,gain}$ and generate the $\log_2$ ratio *.3261* with the likelihood $b_{gain}(.3261)$. Similarly, we can compute the probability of each of the possible state sequences of length four, assuming the fixed observation sequence *Y*. The calculation of $P(Y \mid \lambda)$ according to its direct definition requires 256 state sequences. This calculation is computationally infeasible for large data sets.

The forward algorithm (Baum et al., 1970) is an efficient procedure for the calculation of $P(Y \mid \lambda)$. The forward variable $\alpha_t(i)$ is defined as $\alpha_t(i) = P(y_1, y_2, ..., y_t, q_t = i \mid \lambda)$. It is the probability of the observations $y_1, y_2, ..., y_t$ and being in state *i* at time *t*. The Forward algorithm computes $\alpha_t(i)$ with the following procedure (Rabiner and Juang, 1993):

1. Initialization

$$\alpha_1(i) = \pi_i b_i(y_1), \quad 1 \le i \le n$$

2. Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{n} \alpha_t(i) \cdot a_{ij} \right] \cdot b_j(y_{t+1}), \quad \begin{matrix} 1 \le t \le T-1 \\ 1 \le j \le n \end{matrix}$$

3. Termination

$$P(Y \mid \lambda) = \sum_{i=1}^{n} \alpha_T(i)$$

The Stochastic Forward-Backward algorithm of Bayesian HMM (Guha et al., 2008) begins with the implementation of the Forward algorithm. Figure 4 illustrates a trellis diagram for the calculation of the Forward algorithm. A trellis diagram can be used to visualize likelihood calculations of HMMs. Each column in the trellis shows the possible copy number states at a certain time *t* for four time points. Each state in one column is connected to each state in the adjacent columns by the transition probabilities given by the elements $a_{i,j}$ of the transition matrix *A*. At the bottom is the observation sequence $Y = \{-.3132, -.0855, .0346, .3261\}$.

In the Forward step (Figure 4), at each time-step t, ($t = 1,..,4$) there are four possible CNV states. The algorithm begins with initialization. At the first timestep the prior probability of being in state s_i is multiplied by likelihood of the first $\log_2$ ratio, given the HMM is in the state s_i. The boxes in the first column of Figure 4 give the results of the initialization step. The initial calculations are shown below.

$$\alpha_1(loss) = \pi_{loss} \cdot b_{loss}(-.3132) = (.16)(.96) = .1536$$
$$\alpha_1(normal) = \pi_{normal} \cdot b_{normal}(-.3132) = (.28)(.66) = .1848$$
$$\alpha_1(gain) = \pi_{gain} \cdot b_{gain}(-.3132) = (.27)(.27) = .0729$$
$$\alpha_1(amplification) = \pi_{amplification} \cdot b_{amplification}(-.3132) = (.29)(.03) = .0087$$

Figure 4 also illustrates the induction step for computing the values in the remaining cells of the trellis. At each state (for example the gain state at $t = 2$), there are 4 possible transitions (lines) that reach this state. Elements from the transition and emission matrices *A* and *B* are shown on the transition lines of the trellis of Figure 4. The probability of observations y_1, y_2 and being in the gain state at time $t = 2$, given our

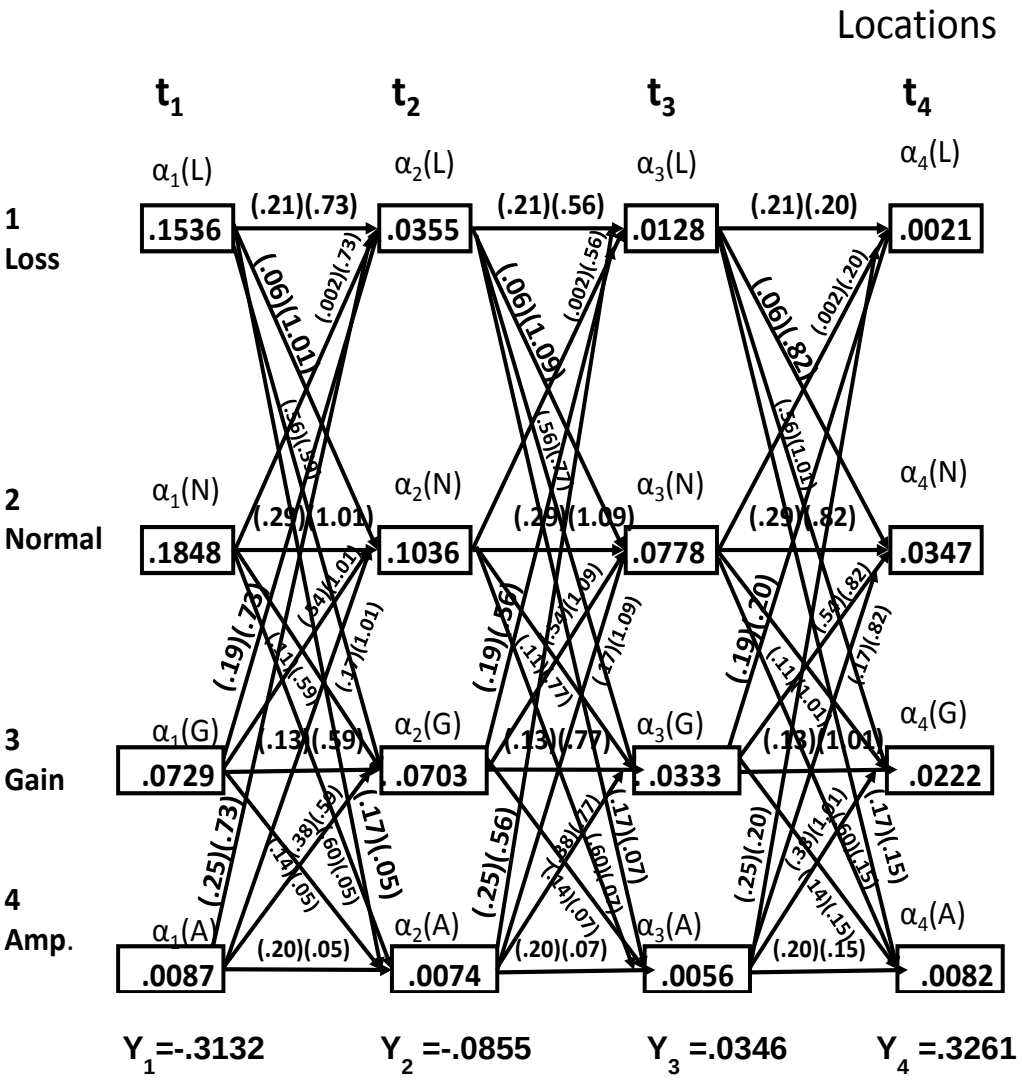
HMM is defined as $\alpha_2(gain)$. The figure shows how the gain state can be reached at time $t = 2$ from the 4 possible states at time $t = 1$. The sum of the probabilities of these 4 transitions is the probability of reaching the gain state at $t = 2$, $\alpha_2(gain)$.

$$\alpha_2(gain) = P(y_1, y_2, q_2 = gain \mid \lambda)$$
$$\alpha_2(gain) = P(-.3132, -.0855, q_2 = gain \mid \lambda)$$
$$\begin{aligned}\alpha_2(gain) = {} & P(\{-.3132, -.0855\}, \{loss, gain\} \mid \lambda) \\ & + P(\{-.3132, -.0855\}, \{normal, gain\} \mid \lambda) \\ & + P(\{-.3132, -.0855\}, \{gain, gain\} \mid \lambda) \\ & + P(\{-.3132, -.0855\}, \{amp, gain\} \mid \lambda)\end{aligned}$$
$$\alpha_2(gain) = (.1536)(.56)(.59) + (.1848)(.11)(.59) + (.0729)(.13)(.59) + (.0087)(.38)(.59)$$
$$\alpha_2(gain) = .0507 + .0120 + .0056 + .0020$$
$$\alpha_2(gain) = .0703$$

This calculation is done at each state for all time-steps, and the results are shown in the boxes of Figure 4. The probabilities of obtaining the loss, normal, gain, and amplification states at $t = 4$ are respectively .0021, .0347, .0222, and .0082. Finally, the last step of the algorithm gives the desired calculation of $P(Y \mid \lambda)$ as the sum of the forward variables $\alpha_4(i)$.

$$P(Y \mid \lambda) = P(\{-.3132, -.0855, .0346, .3261\} \mid \lambda)$$
$$\begin{aligned}P(Y \mid \lambda) = {} & \alpha_4(loss) = .0021 \\ & + \alpha_4(normal) = .0347 \\ & + \alpha_4(gain) = .0222 \\ & + \alpha_4(amplification) = .0082\end{aligned}$$
$$P(Y \mid \lambda) = .0672$$

Figure 4 Forward-Backward sampling- Forward Step. An example I have developed of a continuous HMM with four states. The Forward variables, the sums of the probabilities over all possible paths to any given state are given in the boxes. For example, the probability of reaching the gain state at t_2 is (.1536)(.56)(.59)+(.1848)(.11)(.59)+(.0729)(.13)(.59)+(.0087)(.38)(.59) = .0703.



In the backward step, the final hidden state is sampled based on the forward variables of the last time-step, $\alpha_4(i)$. The algorithm samples the posterior probability distribution of being in the i^{th} state at the last time-step, given the observation sequence and the model.

$$P(S_4 = i \mid y_1, y_2, y_3, y_4) = \frac{P(S_4 = i, y_1, y_2, y_3, y_4)}{P(y_1, y_2, y_3, y_4)}$$

$$P(S_4 = i \mid -.3132, -.0855, .0346, .3261) = \frac{P(S_4 = i, -.3132, -.0855, .0346, .3261)}{P(-.3132, -.0855, .0346, .3261)}$$

$$P(S_4 = i \mid -.3132, -.0855, .0346, .3261) = \frac{\alpha_4(i)}{.0672}$$

The posterior distribution is shown in Table 1. The first MCMC chain indicates the normal state (*2*) as the final hidden state of the sequence based on MAP (maximum a posterior) classification.

Table 1 Forward-Backward sampling- Backward Step
The posterior probability distribution of being in state *i* at location t_4 given the observation sequence.

$S_4 = i$	$P(S_4 = i \mid y_1, y_2, y_3, y_4)$
1	.0021|.0672=.03
2	**.0347|.0672=.52**
3	.0222|.0672=.33
4	.0082|.0672=.12

After sampling S_4 from $P(S_4 = i \mid y_1, y_2, y_3, y_4)$, State *3* is then sampled recursively from $P(S_3 = i \mid S_4 = normal, y_1, y_2, y_3, y_4)$. This is the posterior distribution of being in state *i* at location t_3, given the last hidden state is normal and the data;

$$
\begin{aligned}
P(S_3 = i \mid S_4 = normal, y_1, y_2, y_3, y_4) &\propto P(S_3 = i, S_4 = normal, y_1, y_2, y_3, y_4) \\
&= P(S_3 = i, y_1, y_2, y_3) P(y_4, S_4 = normal \mid S_3 = i, y_1, y_2, y_3) \\
&= P(S_3 = i, y_1, y_2, y_3) P(y_4 \mid S_4 = normal) P(S_4 = normal \mid S_3 = i) \\
&= \alpha_3(i) \cdot b_{normal}(y_4) \cdot a_{i,normal}
\end{aligned}
$$

$$
\begin{bmatrix}
P(S_3 = loss \mid S_4 = normal, Y) \\
P(S_3 = normal \mid S_4 = normal, Y) \\
P(S_3 = gain \mid S_4 = normal, Y) \\
P(S_3 = amplification \mid S_4 = normal, Y)
\end{bmatrix}
\propto
\begin{bmatrix} .0128 \\ .0778 \\ .0333 \\ .0056 \end{bmatrix}
\begin{bmatrix} .82 \end{bmatrix}
\begin{bmatrix} .06 \\ .29 \\ .54 \\ .17 \end{bmatrix}
\propto
\begin{bmatrix} .0173 \\ .5347 \\ .4249 \\ .0231 \end{bmatrix}
$$

The normal state is again selected, now as the third hidden state from the posterior distribution. This process is repeated for state *2* and state *1*, completing the hidden state sequence for the first MCMC iteration. The first row of Table 2 gives the remaining results for the first two states of the first MCMC iteration. The following rows give the results for four additional iterations. The best sequence of copy number assignments for each iteration is given by $Q = \{q_t\}_{t=1}^{4}$. The relative frequency of each event in the sample is shown in Table 3. These frequencies represent the MCMC posterior probability of each state. The hidden state labels are {*loss*, *normal*, *normal*, *gain*} for {State *1*, State *2*, State *3*, State *4*} respectively. This process extended to n genetic positions in the algorithm.

Table 2 Forward-Backward sampling- Posterior Samples
The state labels for each clone from five MCMC iterations

$\mathbf{q_1}$	$\mathbf{q_2}$	$\mathbf{q_3}$	$\mathbf{q_4}$
$\mathbf{q_1^{(1)} = 2}$	$\mathbf{q_2^{(1)} = 2}$	$\mathbf{q_3^{(1)} = 2}$	$\mathbf{q_4^{(1)} = 2}$
$q_1^{(2)} = 1$	$q_2^{(2)} = 2$	$q_3^{(2)} = 3$	$q_4^{(2)} = 3$
$q_1^{(3)} = 1$	$q_2^{(3)} = 2$	$q_3^{(3)} = 3$	$q_4^{(3)} = 3$
$q_1^{(4)} = 1$	$q_2^{(4)} = 2$	$q_3^{(4)} = 2$	$q_4^{(4)} = 2$
$q_1^{(5)} = 2$	$q_2^{(5)} = 1$	$q_3^{(5)} = 2$	$q_4^{(5)} = 3$

Table 3 Forward-Backward sampling- Posterior Inference
The hidden state estimates at each location are shown in bold. The relative frequency of each event in the sample is used to approximate the posterior probability of each hidden state.

$\mathbf{q_1}$	$\mathbf{P(q_1=s_i)}$	$\mathbf{q_2}$	$\mathbf{P(q_2=s_i)}$	$\mathbf{q_3}$	$\mathbf{P(q_3=s_i)}$	$\mathbf{q_4}$	$\mathbf{P(q_4=s_i)}$
1	**3|5**	1	1|5	1	0	1	0
2	2|5	**2**	**4|5**	**2**	**3|5**	2	2|5
3	0	3	0	3	2|5	**3**	**3|5**
4	0	4	0	4	0	4	0

Chapter 3

Assessment of the Programs

3.1 Methods Considered

Numerous algorithms exist for array CGH data analysis. Rueda and Diaz-Uriarte (2007) developed Reversible Jump Array Comparative Genomic Hybridization (RJaCGH). Guha et al. (2008) created Bayesian Hidden Markov Modeling of array CGH data (Bayesian HMM). Engler et al. (2006) generated a pseudolikelihood approach for simultaneous analysis of array comparative genomic hybridizations (CGHclassify). Marioni et al. (2006) developed a heterogeneous hidden Markov model for segmenting array CGH data (BioHMM). Shah et al. (2006) integrated copy number polymorphisms into array CGH analysis using a robust HMM (HMM-R). All five packages are publicly available and output estimated aberrations in copy number.

3.1.1 The RJaCGH Package

The first program (Rueda and Diaz-Uriarte, 2007) considers the space between clones, and provides probabilities of a copy number change instead of p-values or segment means. The software allows for either chromosome specific or genome-wide analysis. RJaCGH applies a non-homogeneous HMM. The transition probability matrix is not constant in time but varies with the distance between genes, thus the adjective non-homogeneous. Space between genes is an important consideration in the model, taking into account the variable nature in which probes are separated on the array. The hidden states are derived from executions of a reverse jump Markov Chain Monte Carlo (RJMCMC). The RJMCMC algorithm proposed by Green (1995) provides a powerful

tool to the Bayesian model determination problem. Since the method allows for jumps between states of differing dimensions, many models may be visited (Brooks et al., 2003). Results are finally summarized by Bayesian Model Averaging, incorporating model uncertainty (Rueda and Diaz-Uriarte, 2007).

3.1.2 The BioHMM Package

In brief, BioHMM (Marioni et al., 2006) incorporates a HMM where the optimal segmentation of clones is found by likelihood maximization using a derived number of Gaussian Distributions with state-specific means and fixed variance. Akaike Information Criterion (AIC) is the criterion for model selection, used here to select the (*k*) states. Partitioning around medoids (PAM) is a widely used partitioning method that divides the data into a prespecified number of mutually exclusive groups. PAM segments the observations in *k* states, and allows for the approximation of each state's mean. The Viterbi algorithm (Viterbi, 1967) is used to find the most probable sequence of hidden states given a sequence of intensity ratios.

3.1.3 The CGHclassify Package

The third program (Engler et al., 2006) uses a 3-state Gaussian mixture HMM that exploits features that are shared in common among chromosomes. The method introduces spatial dependence by using moving windows of three intensities. The authors use a pseudolikelihood function as the basis for estimation. The most probable copy number assignments are inferred from the posterior probabilities of the states.

3.1.4 The Bayesian HMM Package

The fourth program (Guha et al., 2008) applies a Bayesian HMM that uses a hybrid MCMC algorithm to obtain samples of the parameters. Guha et al. (2008) assume a model of K=4 states, where s_1 corresponds to a copy number loss, s_2 corresponds to the normal state, s_3 to single copy gain, and s_4 to amplification. Conditional on the copy number states, the normalized $\log_2$ ratios are assumed to have Gaussian distributions with state-specific means and variance. The Bayesian approach assumes priors for all unknown parameters. Priors for the parameters of the emission distribution were chosen based on theoretical values. Gibbs sampling is used for sampling the parameters of the emission matrix. A Metropolis-Hastings algorithm is used to sample the parameters of the transition matrix. A stochastic version of the forward-backward algorithm determines the hidden states (Chib, 1996).

3.1.5 The HMM-R Package

Shah et al. (2006) modified Bayesian HMM to take into account outliers. The authors considered the data as belonging to one of the states from the set $\{Loss, Normal, Gain, Amplification\}$. These states are called "inlier" states. The $\log_2$ ratio expected from an outlier can be the result of measurement noise or the mislabeling of clones. A robust HMM (HMM-R) was proposed by Shah et al. (2006). The emission pdf is a mixture of two normal densities, one component represents inlier clones and the other represents outliers. The emission pdf is modified as follows:

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

where x_i is the $\log_2$ ratio for clone i, $S_i = s$ is the state label at i, and $O_i = 1$ means location i is an outlier.

3.2 Simulation Settings

For this analysis, I used the simulated data set from Willenbrock and Fridlyand (2005) created for an earlier comparative study of segmentation approaches (called WF Data from now on). Andersson et al. (2008), Wang and Wang (2007), Nguyen et al. (2006), Rueda and Diaz-Uriarte (2007), Shah et al. (2006), Van de Wiel et al. (2007), and others, demonstrate the capabilities of their techniques by applying them to the WF Data. The WF Data present a suitably varied set necessary for accurate assessment of these distinct algorithms. This is data simulated to emulate the complexity of real tumor profiles and designed to become a standard for systematic comparisons of computational segmentation approaches (Fridlyand et al., 2004). The advantages of using synthetic data are two-fold. First, the ground truth locations of the aberrations are known. Second, we can control the difficulty of the problem. The WF Data is considerably harder (but more realistic) than other synthetic datasets used in earlier papers (Shah et al., 2006). I downloaded the data from http://www.cbs.dtu.dk/~hanni/aCGH, and the specific file used was http://www.cbs.dtu.dk/~hanni/aCGH.simulated.data.RDATA.

I ran the five software packages on all replicates of the WF Data. The WF Data consists of 500 replicates, each with 2,000 clones from 20 chromosomes; that is, 100 clones per chromosome. The WF Data consists of expected $\log_2$-ratios of each clone; that is, the CNV and the simulated $\log_2$ of tumor intensity over reference intensity, measured on a microarray. The start and end position measured in base pairs of each clone are given. The length of each clone is 1,000 base pairs. They are sequentially

ordered along the first 100kb of each chromosome. The chromosomal segments with DNA copy number c = 0, 1, 2, 3, 4 and 5 are generated with probability 0.006, 0.053 0.872, 0.047, 0.015 and 0.007. Table 4 shows the first 15 simulated $\log_2$ ratios for the first synthetic sample.

Table 4. First 15 clones on chromosome 1 from Sample 1

sample1	Chrom	log2ratios	copynumber	gain.loss	kb	index
1	1	-0.0159371	2	0	1	1
2	1	-0.2857084	2	0	2	2
3	1	-0.2563104	2	0	3	3
4	1	-0.1095165	2	0	4	4
5	1	0.07679709	2	0	5	5
6	1	0.10197522	2	0	6	6
7	1	0.18096776	2	0	7	7
8	1	-0.3464299	2	0	8	8
9	1	-0.2376993	2	0	9	9
10	1	0.06812582	2	0	10	10
11	1	0.02651082	2	0	11	11
12	1	-0.0411707	2	0	12	12
13	1	-0.2079212	2	0	13	13
14	1	0.23193684	2	0	14	14
15	1	-0.2152985	2	0	15	15

The output of each method includes the estimated copy number states. Following Willenbrock and Fridlyand (2005), I reduced events to one of three categories: deletion, normal, and gain. The HMM-R and Bayesian HMM output differentiates four states: single-gains, amplifications, as well as normal and loss. Therefore, in the post-processing of HMM output, I combined single gain and amplification as a gain. CGHclassify and BioHMM do not allow for states of multiple-gains. In the RJaCGH usage argument, I set the maximum number of hidden states to three.

3.3 Software Methods

With regard to software implementation, two platforms were required to execute the algorithms. Bayesian HMM (Guha et al., 2008) and HMM-R (Shah et al., 2006) were

implemented in Matlab®. Bayesian HMM required an additional purchase of the Bioinformatics Toolbox 3.3 (2009a, The MathWorks, Inc., USA). The Toolbox also allows you to perform chromosomal segmentation with the CBS algorithm (Olshen et al., 2004), and follows this with the capability for analysis with Bayesian HMM. CGHclassify (Engler et al., 2006) and RJaCGH (Rueda and Diaz-Uriarte, 2007) were both executed on a freely available R system. The HMMs were run with their default parameters. Table 5 (Shah et al., 2006) provides the default parameter settings for the software packages.

Table 5a Summary of parameter settings for Bayesian HMM

Parameter	**Description**	**Value**
δ	Dirichlet prior on the i^{th} row of transition matrix *A*	1, 1, 1, 1
$\alpha_{1:4}$	Shape of gamma prior on σ^{-2}	1, 1, 1, 1
$\beta_{1:4}$	Scale of gamma prior on σ^{-2}	1, 1, 1, 1
$m_{1:4}$	Normal prior mean on means μ	-1, 0, .58, 1
$\tau^2_{1:4}$	Normal prior variance on means μ	1, 1, 1, 4

Note: The programming language used was Matlab® version 7.8 (R2009a).

Source: Shah et al. 2006

Table 5b Summary of parameter settings for HMM-R

Parameter	Description	Value
δ	Dirichlet prior on the i^{th} row of transition matrix *A*	1, 1, 1, 1
$\alpha_{1:4}$	Shape of gamma prior on σ^{-2}	10, 100, 5, 5
$\beta_{1:4}$	Scale of gamma prior on σ^{-2}	1, 1, 1, 1
$m_{1:4}$	Normal prior mean on means μ	-1, 0, .58, 1
$\tau^2_{1:4}$	Normal prior variance on means μ	.5, .001, 1, 1

Note: The programming language used was Matlab® version 7.8 (R2009a).

Table 5c Summary of parameter settings for RJaCGH

Parameter	Description	Value
$m_{1:3}$	Normal prior mean on means	Median (y)
$\tau^2_{1:3}$	Normal prior variance on means	Range (y)
$\alpha_{1:3}$	Shape of gamma prior on σ^{-2}	2
$\beta_{1:3}$	Scale of gamma prior on σ^{-2}	$Range^2$ (y)/50
Beta	The model for the transition matrix is based on Beta	Gamma(1, 1)

Note: The programming language used was R version 2.6.2.

Table 5d Summary of parameter settings for CGHclassify

Parameter	Description	Value
mu_L	Mean of copy number loss state	-.5
m0	Mean of copy number no-change state	0
mu_G	Mean of Copy number gain state	.5
s2	Variance of log2 ratios in each state	.1
p1	Probability of gain across entire data set	.1
pn1	Probability of loss across entire data set	.1
p10	Transition probability from gain to no-change	.1
pn11	Transition probability from loss to gain	.1
p0n1	Transition probability from no-change to loss	.1

Note: The programming language used was R version 2.6.2.

For CGHclassify, an error of "non-finite value" may be caused by an inadequate parameterization of the defaults. Engler et al. (2006) suggest increasing the starting *s2* parameter to 0.2 and rerunning the analysis to remedy this. In the event this fails, no further recommendations have been specified. When this occurred, I increased the *s2* parameter as recommended and report the number of failed analyses.

The BioHMM approach does not prespecify the underlying copy number events on a given chromosome, but rather focuses on the identification of segments of common $\log_2$ ratio mean. Thus the states in the Fridlyand et al. (2004) approach are not underlying copy number events such as gain and loss, but are segments of common mean. A change in state corresponds to a breakpoint. Following segmentation, genetic features such as focal aberrations and amplifications (low- and high-level alteration within a segment involving a small number of clones) are identified (Engler et al., 2006). The software's output needs to be further analyzed in order to "call" the gains and losses. Willenbrock and Fridlyand (2005) developed the MergeLevels algorithm to determine which segments of common mean represent real genetic alterations. MergeLevels reduces the number of segments by merging ones that are likely to correspond to the same copy number. Following segment combination, the segment level with predicted $\log_2$ ratio closest to 0, which is also the level with the largest number of observations, is assigned to the "normal" class. The remaining levels are assigned to either "gain" or "loss" depending on whether their mean $\log_2$ ratio was larger or smaller, respectively, than the "normal" class.

I analyzed the performance of the BioHMM method with the program Analysis of Data from aCGH experiments (Diaz-Uriarte and Rueda, 2007). ADaCGH is both an R package and a web-based application for the analysis of aCGH data. The program implements eight methods for detection of CNVs, including BioHMM by Marioni et al. (2006). Computational efficiency and decreased user wait time are benefits to working within the ADaCGH platform (Diaz-Uriarte and Rueda, 2007). I ran aCGH and post-processed the MergeLevels output using the ADaCGH web-based tool available from the website http://adacgh2.bioinfo.cnio.es.

3.4 Performance Measures

I define classification accuracy as the percentage of clones for which the classification agreed with the known categories. A clone whose fitted state differs from the simulated ground truth label, is defined to be an error. Classification accuracy, false discovery rate, specificity, and sensitivity were used as performance metrics. Following Rueda and Diaz-Uriarte (2007), I used the confusion matrix defined in Table 6 to estimate rates. I also calculated Cohen's Kappa coefficient (Cohen, 1960) from the confusion matrices to measure agreement beyond chance between the fitted results and the ground truth data. Kappa values range between -1 (all clones incorrectly fitted) and 1 (all clones correctly fitted). A Kappa value equal to zero indicates a performance no better than random.

Table 6 A confusion matrix (Provost and Kohavi, 1998) contains information about actual and predicted classifications derived by a classification system. The confusion matrix used to calculate rates (Rueda and Diaz-Uriarte, 2007).

		Fitted State			
		gain (g)	**n**ormal (n)	**l**oss (l)	Total
True State	**G**ain (G)	Gg	Gn	Gl	G.
	Normal (N)	Ng	Nn	Nl	N.
	Loss (L)	Lg	Ln	Ll	L.

Correct Classification rate

$$CCR = \frac{Gg + Nn + Ll}{G. + N. + L.}$$

False Discovery Rate

$$FDR = \frac{Ng + Nl}{Gg + Ng + Lg + Gl + Nl + Ll}$$

Specificity

$$specificity = \frac{Nn}{Ng + Nn + Nl}$$

Sensitivity

$$sensitivity = \frac{Gg + Ll}{Gg + Gn + Gl + Lg + Ln + Ll}$$

Kappa

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

Additionally, I report the classification error rates as follows:

$$\varepsilon_{12} = \hat{P}(\text{Gain incorrectly fitted as Normal}) = \hat{P}(\text{Normal state observed} \mid \text{Gain true state})$$
$$\varepsilon_{13} = \hat{P}(\text{Gain incorrectly fitted as Loss}) = \hat{P}(\text{Loss state observed} \mid \text{Gain true state})$$
$$\varepsilon_{21} = \hat{P}(\text{Normal incorrectly fitted as Gain}) = \hat{P}(\text{Gain state observed} \mid \text{Normal true state})$$
$$\varepsilon_{23} = \hat{P}(\text{Normal incorrectly fitted as Loss}) = \hat{P}(\text{Loss state observed} \mid \text{Normal true state})$$
$$\varepsilon_{31} = \hat{P}(\text{Loss incorrectly fitted as Gain}) = \hat{P}(\text{Gain state observed} \mid \text{Loss true state})$$
$$\varepsilon_{32} = \hat{P}(\text{Loss incorrectly fitted as Normal}) = \hat{P}(\text{Normal state observed} \mid \text{Loss true state})$$

Table 7 allows us to visualize the misclassification costs of calling a clone incorrectly (Kim, Gordon, Sebat, Ye, and Finch, 2008). Table 7 presents the estimated conditional probability of a fitted state, given the true state.

Table 7 Estimated conditional probability of a fitted state, given true state*

	Fitted State		
True State	**g**ain	**n**ormal	**l**oss
Gain	**$1-\varepsilon_{12}-\varepsilon_{13}$**	**ε_{12}**	**ε_{13}**
Normal	**ε_{21}**	**$1-\varepsilon_{21}-\varepsilon_{23}$**	**ε_{23}**
Loss	**ε_{31}**	**ε_{32}**	**$1-\varepsilon_{31}-\varepsilon_{32}$**

* *ij*th cell is defined as a function of error model parameters ε_{ij} (where *i, j* are one of Gain, Normal, or Loss). As defined in Methods, these parameters are:

ε_{12} = $\hat{P}$ (Gain incorrectly fitted as Normal)
ε_{13} = $\hat{P}$ (Gain incorrectly fitted as Loss)
ε_{21} = $\hat{P}$ (Normal incorrectly fitted as Gain)
ε_{23} = $\hat{P}$ (Normal incorrectly fitted as Loss)
ε_{31} = $\hat{P}$ (Loss incorrectly fitted as Gain)
ε_{32} = $\hat{P}$ (Loss incorrectly fitted as Normal)

Chapter 4

Results

4.1 Detection of candidate copy number variants from simulated data

The graphs of figure 5 are comparable to the plots of Shah et al. (2006). Figure 5 displays chromosome 4 of sample 28 with the true classifications (Panel A), the fitted states of RJaCGH (Panel B), CGHClassify (Panel C), Bayesian HMM (Panel D), BioHMM (Panel E) and HMM-R (Panel F). The first 18,000 base pairs (bp) on chromosome 4 is a segment of loss. This segment is followed by a 20,000 bp region of gain. The remaining length of the chromosome retains a normal copy number. All five of the programs accurately identify all of the loss CNVs. Each program also correctly calls all normal $\log_2$-ratios without any false discoveries. However, each program incorrectly calls normal states in the middle of the gain segment. BioHMM has the lowest accuracy call rate for gain, with zero labeled correctly. HMM-R has the highest accuracy, correctly labeling 75% of the gain CNVs. The remaining packages have accuracy call rates under 25% for this region on chromosome 4.

Figure 5 Panel *A* depicts chromosome 4 of sample 28. The chromosomal position in base pairs (bp) is plotted on the horizontal axis. Clones associated with normal $\log_2$-ratios are plotted in blue, losses in red, and gains in green. The vertical lines in the remaining graphs represent states fitted by the software.

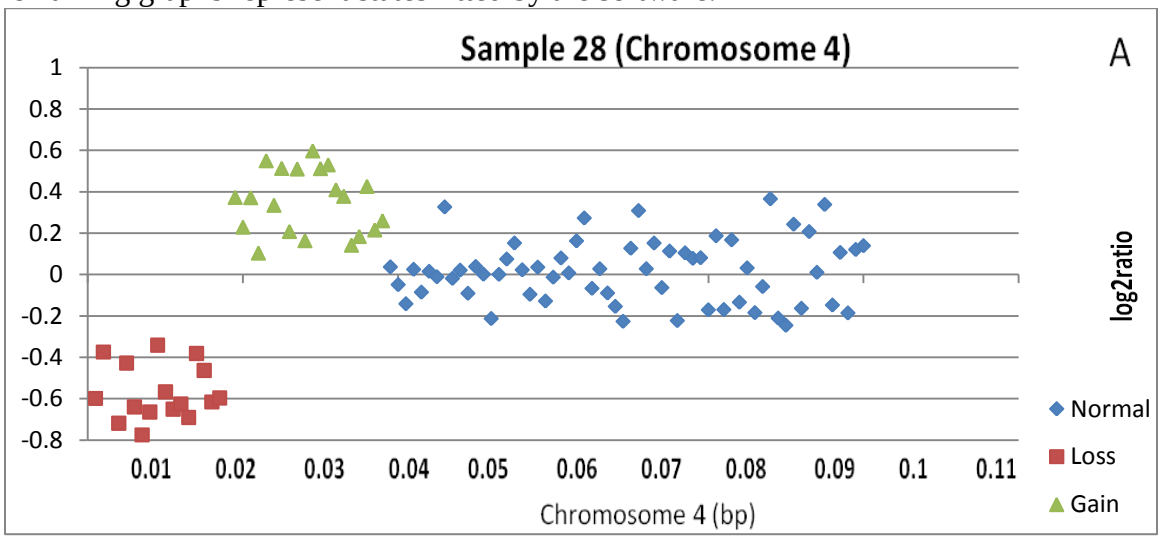


Figure 5 In Panel *B*, RJaCGH commits seventeen errors. Black vertical lines represent the errors produced by the software. The black lines correspond to 17 uncalled gains.

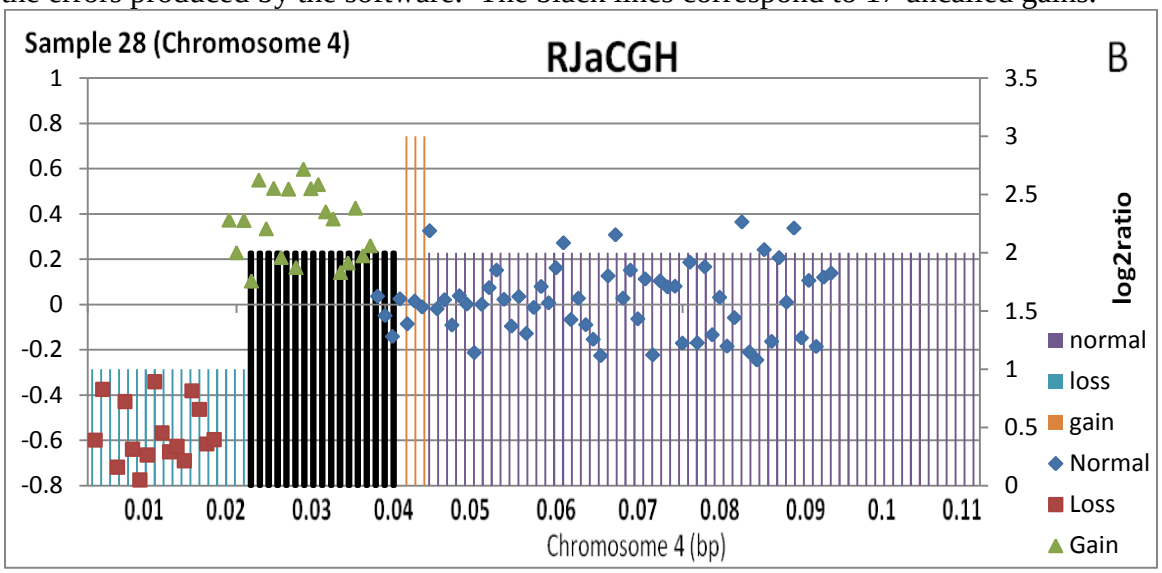


Figure 5 In Panel *C*, CGHclassify commits seventeen errors. The black lines of Panel *C* are indicative of 16 gains that are mislabeled as normal by CghClassify.

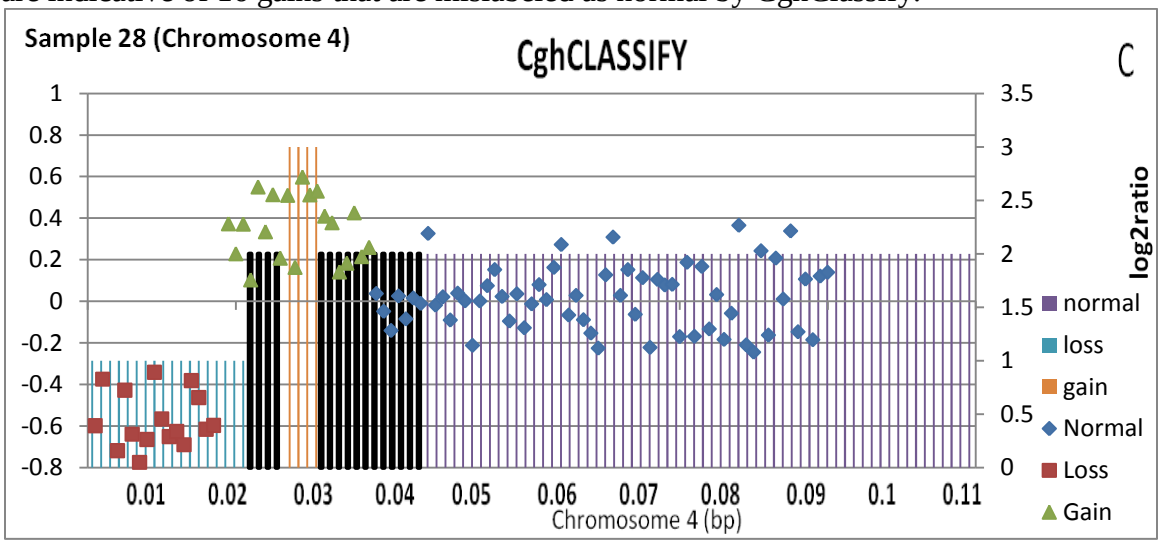


Figure 5 In Panel *D*, BayesianHMM commits seventeen errors. The black lines of Panel *D* are indicative of 19 gains that are mislabeled as normal by BayesianHMM.

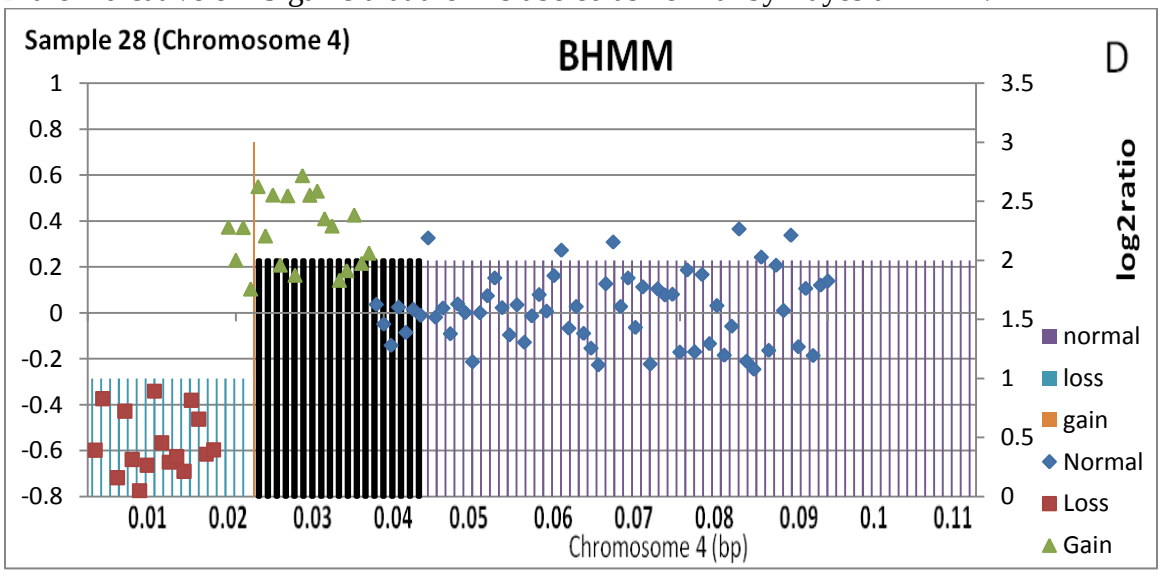


Figure 5 In Panel *E*, BioHMM commits 20 errors. BioHMM misses the entire gain segment in Panel *E*.

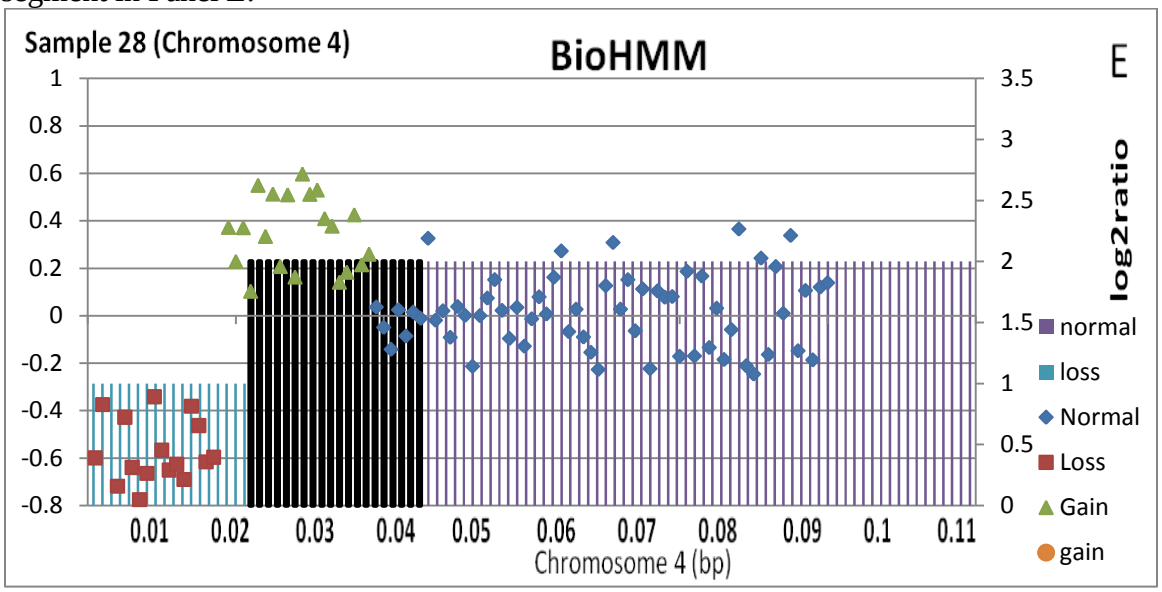
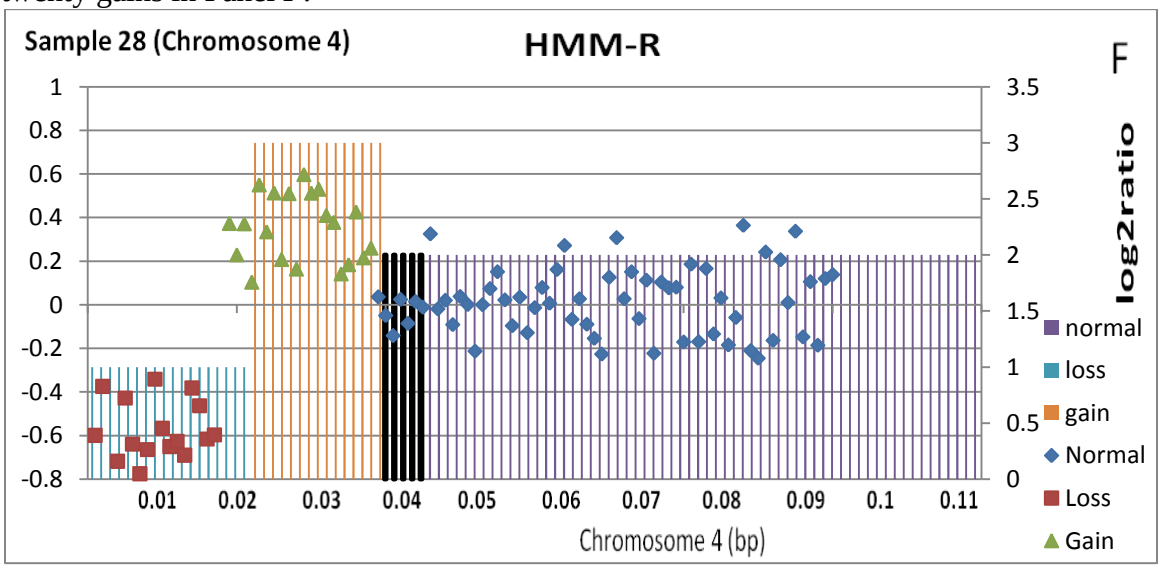


Figure 5 In Panel *F*, HMM-R commits 5 errors. HMM-R correctly labels fifteen of the twenty gains in Panel *F*.



The total number of CNVs detected by each of the five software packages is given in Table 8. I also include the number of CNVs that were correctly identified (True CNVs). The true positive rates fell between .65 and .87. The true positive rate for CGHclassify was calculated on 93% of the data sets. I was unable to produce results for thirty-three samples.

Bayesian HMM more effectively detected gains compared to detecting losses. CGHclassify, HMM-R, BioHMM and RJaCGH detected deletions and duplications with similar percentages.

Table 8 Number of candidate aberrations from the simulated data. True CNVs represent correctly fitted variants. The simulated data set contained 128,299 gains and losses.

Proram	**#CNVs**	**#Gains**	**%Gains**	**#Losses**	**%Losses**	**# True CNVs**	**True Positive Rate** *
BHMM	151,409	89,483	59.1	61,926	40.9	100,544	.78
CGH classify	85,410	42,921	50.3	42,489	49.7	83,665	.65
RJaCGH	93,054	46,534	50.0	46,520	50.0	91,312	.71
BioHMM	88,447	45,803	51.8	42,644	48.2	85,957	.67
HMM-R	113,502	58,665	51.7	54,837	48.3	111,520	.87
WF Data	**128,299**	**68,333**	**53.3**	**59,896**	**46.7**	**128,299**	**1.00**

* The true positive rate for CghClassify was calculated on 93% of the data sets. I was unable to process thirty-three samples. The unprocessed datasets comprised 8, 861 CNVs, including 55% gains, and 45% losses.

4.2 Performance statistics for predicting gains and losses

The processing results of each algorithm are given in Tables 9-13. I determined the entries of the confusion matrix as outlined in Table 6 of Methods. The entries in the confusion matrix have the following meaning in the context of this study: *Gg* is the

number of correct predictions that an instance is a gain. *Gn* is the number of incorrect predictions that an instance is normal when in fact it is a gain. *Gl* is the number of incorrect predictions an instance is a loss when it is a gain.

Table 9. RJaCGH Confusion Matrix

True State	**Fitted**				
		g	**n**	**l**	**Total**
	G	45,570	22,809	11	**68,390**
	N	958	870,080	767	**871,805**
	L	6	14,057	45,742	**59,805**
	Total	**46,534**	**906,946**	**46,520**	**1,000,000**

Table 10. CGHclassify Confusion Matrix

True State	**Fitted**				
		g	**n**	**l**	**Total**
	G	41,931	21,343	11	**63,285**
	N	984	813,074	744	**814,802**
	L	6	14,173	41,734	**55,913**
	Total	**42,921**	**848,590**	**42,489**	**934,000**

Table 11. Bayesian HMM Confusion Matrix

True State	Fitted				
		g	**n**	**l**	**Total**
	G	50,907	17,321	93	**68,321**
	N	38,433	821,280	12,196	**871,909**
	L	143	9,990	49,637	**59,770**
	Total	**89,483**	**848,591**	**61,926**	**1,000,000**

Table 12. BioHMM Confusion Matrix

True State	Fitted				
		g	**n**	**l**	**Total**
	G	44,484	23,771	4	**68,259**
	N	1,319	869,534	1,167	**872,020**
	L	0	18,248	41,473	**59,721**
	Total	**45,803**	**911,553**	**42,644**	**1,000,000**

Table 13. HMM-R Confusion Matrix

True State	Fitted				
		g	**n**	**l**	**Total**
	G	57,668	10,574	17	**68,259**
	N	842	870,210	968	**872,020**
	L	155	5,714	53,852	**59,721**
	Total	**58,665**	**886,498**	**54,837**	**1,000,000**

Table 14 summarizes the four performance statistics for predicting gains and losses in the synthetic data set. HMM-R achieved a correct classification accuracy of 98.2%, higher than any of the other methods. Here RJaCGH, CGHclassify and BioHMM have essentially equal classification accuracy rate of 96%. Bayesian HMM had the worst correct classification rate at 92.2%. HMM-R had a false discovery rate of 1.6%. RJaCGH and CGHclassify also had a false discovery rate almost as good. HMM-R had the largest sensitivity at over 87%. RJaCGH, CGHclassify, and HMM-R achieved a specificity of 99.8%. The Cohen's Kappa coefficients range from .688 to .917. The Kappa values indicate that model results were not due to chance.

Table 14 The performance statistics for the compared methods on the synthetic data, and Cohen's Kappa coefficient.

Method	BHMM	RJaCGH	CGHclassify *	BioHMM	HMM-R
Correct Classification Rate	.922	.961	.960	.955	.982
False Discovery Rate	.334	.019	.020	.028	.016
Specificity	.942	.998	.998	.997	.998
Sensitiviy	.785	.712	.702	.672	.871
Cohen's Kappa coefficient	.688	.810	.802	.777	.917

* The rates for CGHclassify were calculated on 93% of the data sets.

4.3 Classification error rates

I determined the classification error rates as outlined in Table 7 of Methods. RJaCGH, CGHclassify, and HMM-R called truly normal clones with 99.8% accuracy. The error rate in classifying a gain as normal was the lowest for HMM-R at around 16%. RJaCGH, CGHclassify, and BioHMM were almost identical with this error rate around 33%. These values are displayed in Table 15 below. Error for true loss was also the lowest with HMM-R, and the highest with BioHMM. The error rates ranged from 9.6% to 30.6%. Incorrectly calling a gain a loss, or a loss a gain, is the most uncommon error for this data set.

Table 15 Estimated conditional probability of a fitted state, given true state. Where $\hat{P}(state_j \text{ observed} \mid state_i \text{ true}) = \varepsilon_{ij}$

BHMM $\varepsilon = \begin{pmatrix} .745 & .253 & .001 \\ .044 & .942 & .014 \\ .002 & .167 & .830 \end{pmatrix}$
RJaCGH $\varepsilon = \begin{pmatrix} .666 & .334 & .000 \\ .001 & .998 & .001 \\ .000 & .235 & .765 \end{pmatrix}$
CGHclassify $\varepsilon = \begin{pmatrix} .663 & .337 & .000 \\ .001 & .998 & .001 \\ .000 & .253 & .746 \end{pmatrix}$
BioHMM $\varepsilon = \begin{pmatrix} .652 & .348 & .000 \\ .002 & .997 & .001 \\ .000 & .306 & .694 \end{pmatrix}$
HMM-R $\varepsilon = \begin{pmatrix} .845 & .155 & .000 \\ .001 & .998 & .001 \\ .002 & .096 & .902 \end{pmatrix}$

Chapter 5

Conclusions and Future Studies

5.1 Conclusions

The HMM framework has been criticized for its poor treatment of outliers. The concern for over-segmentation, and segments spanning single clones, has been addressed by Shah et al. (2006). They extended the Bayesian HMM to create a more robust method for handling "outlying clones". In this study the HMM-R program achieved the highest correct classification rate, specificity, and the lowest false discovery rate. Other methods also had lower levels of the true positive rate. Misclassification of the most common category to any other category has been proven to be the most costly error in tests of association in terms of statistical power (Kang et. al, 2004). HMM-R produced the best results for misclassifying truly normal clones. Finally, HMM-R had the highest Kappa value, much higher than RJaCGH or CGHclassify. For this simulated data set HMM-R outperformed the other methods.

RJaCGH (kappa = .81), and CGHclassify (kappa=.80) ranked second and third in this analysis. RJaCGH had the second highest Kappa value, indicating this software had higher chance-corrected agreement with the ground truth data than CGHclassify. The RJaCGH package (Rueda and Diaz-Uriarte, 2007) for fitting the non-homogeneous HMM to aCGH data through Bayesian methods and Reversible Jump Markov chain Monte Carlo was very straightforward to execute. Rueda and Diaz-Uriarte (2007) believe that an appropriate method for array CGH analysis should consider space between clones, should provide probabilities of a copy number change instead of p-values, and should allow for chromosome and genome-wide analysis.

HMM-R, RJaCGH, and CGHclassify had false discovery rates under 2%. These rates are lower than the rates exhibited in previous studies of CBS and the first HMM for copy number detection (Fridlyand, 2004). For CGHclassify, this false discovery rate, however, was calculated on just 93% of the data sets. I was unable to process 7% of the data. The user is not guaranteed output from the default parameter settings. I found the usage guidelines inadequately described a methodology that would ensure obtaining copy number calls.

I have shown that Bayesian HMM and BioHMM were only able to achieve a false discovery rate of .33 with a sensitivity of .67, while the objective studies of CBS (Lai et al., 2005; Willenbrock and Fridlyand, 2005) indicate a false discovery rate of .06, and sensitivity of .88. Additionally, Bayesian HMM and BioHMM misclassified the most common category more frequently than the other three packages. Since each chromosome must be analyzed individually, Bayesian HMM is also computationally burdensome for whole genome analysis. The Bayesian HMM method required that each chromosome of each sample was individually analyzed, unlike the other four programs which offer simultaneous analysis of the genome. BioHMM (kappa = .78), and Bayesian HMM (kappa=.69) ranked fourth and fifth in this analysis.

This study demonstrated that the accuracy rates of HMM software has improved since Fridlyand et al. (2004). I have also shown appreciable error rates in copy number prediction for five current HMMs. Three software packages had error rates for true gains over 33%. The error rate in classifying a gain as normal was around 16% for HMM-R, the best performing software. There is significant room to improve the HMM programs' error rates.

5.2 Future Studies

The goal of DNA sequencing is to determine the exact order of the four nucleotides in a segment of DNA. One approach used to sequence the whole genome is called shotgun sequencing (Venter et al., 1998). The sequence of a DNA segment is produced from a large number of short nucleotide sequences, called reads. The shotgun reads (fragments) are read by automated sequencing machines. Specialized computer programs assemble the reads together into the original genome. The recent introduction of sequencing methods capable of producing millions of reads is rapidly changing the landscape of genetics. High-throughput, also known as next-generation sequencing (NGS) technology, can produce megabases of sequence for a fraction of the price and time of previous technologies (Mardis, 2008). Presently, the leading NGS platforms are 454/Roche (Margulies et al., 2005), ABI SOLID (Shendure et al., 2005) and Illumina/Solexa (Bennett, 2004).

The resolution of array CGH microarrays determines CNV detection effectiveness. Currently available array platforms consisting of more than 1 million probes have a lower limit of detection of 10-25kb (Yoon et al., 2009). CNVs detected from array CGH are larger sized resulting from inadequate resolution. Array-based approaches are also generally too noisy to discern subtle copy number differences (such as 15 copies versus 12 copies). As a consequence, connections between higher-order CNVs and diseases are not detected in genome-wide association studies (Chiang and McCarroll, 2009). Methods to detect CNVs using aCGH approaches now face considerable competition from NGS technology.

Sequencing using next-generation technology has several advantages that make it a potentially powerful alternative to aCGH for identifying genomic variations such as deletions and duplications (Daines et al., 2009). The current cost of CNV discovery by sequencing is comparable to or lower than that of aCGH and is continuing to decline. Sequencing data can also be reprocessed for varied purposes as opposed to data from microarrays that is typically utilized by only a single study (Xie and Tammi, 2009). Additionally, the majority of human CNVs are relatively small, containing less than 10 kb of sequence (Eichler, 2006). NGS based algorithms have demonstrated the feasibility to identify CNVs of variable lengths, including small ones that microarray based programs miss (Yoon et al., 2009). Currently, CNVs are now more precisely and efficiently discovered from NGS data (Chiang and McCarroll, 2009).

Sequence coverage is defined as the average number of times any given genomic base is represented in the sequence reads (Deonier et al., 2005). Variation in sequencing coverage in genome assemblies has been used as an indicator for potential CNV between an assembled genome and sequencing data from another genome (Xie and Tammi, 2009). One attempts to find regions with unusually high or low coverage in the alignments of reads to a reference genome. These regions may represent CNVs. Deletions and duplications are discovered from the depth of coverage (read depth) of mapped reads from NGS platforms.

In the last few years, read depth sequencing strategies for cost-effective genome-wide characterization of CNVs have been developed (e.g., Chiang and McCarroll, 2009; Daines et al., 2009; Sudbery et al., 2009; Yoon et al., 2009; Xie and Tammi, 2009). An unexamined weakness is the programs vulnerability to sequencing errors. There has been

no investigation on the effect of sequencing errors on read depth CNV detection. Small error rates in sequencing have proven to be significant for rare variants (Bravo and Irizarry, 2009). A goal for future studies is to determine the effect of sequencing errors on CNV prediction accuracy.

References

Andersson R, Bruder CEG, Piotrowski A, Menzel U, Nord H, Sandgren J, Hvidsten TR, de Stahl TD, Dumanski JP, Komorowski J: **A segmental maximum a posteriori approach to genome-wide copy number profiling**. *Bioinformatics* 2008, **24**(6):751-758.

Baum L, Petrie T, Soules G, Weiss N: **A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains**. *The Annals of Mathematical Statistics* 1970, **41**(1):164-171.

Bennett S: **Solexa Ltd**. *Pharmacogenomics* 2004, **5**(4):433-438.

Bravo HC, Irizarry RA: **Model-based quality assessment and base-calling for second-generation sequencing data**. *Biometrics* 2009.

Brooks SP, Giudici P, Roberts GO: **Efficient construction of reversible jump markov chain monte carlo proposal distributions**. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 2003, **65**(1):3-39.

Brown CS, Goodwin PC, Sorger PK: **Image metrics in the statistical analysis of DNA microarray data**. *Proceedings of the National Academy of Sciences of the United States of America* 2001, **98**(16):8944-8949.

Chiang D, McCarroll S: **Mapping duplicated sequences**. *Nature Biotechnology* 2009, **27**(11):1001-1002.

Chib S: **Calculating posterior distributions and modal estimates in Markov mixture models**. *Journal of Econometrics* 1996, **75**(1):79-97.

Cohen J: **A coefficient of agreement for nominal scales**. *Educational and Psychological Measurement* 1960, **20**(1):37-46.

Daines B, Wang H, Li Y, Han Y, Gibbs R, Chen R: **High-throughput multiplex sequencing to discover copy number variants in drosophila**. *Genetics* 2009, **182**(4):935-941.

Deonier R, Tavaré S, Waterman M: **Computational genome analysis: An introduction**: Springer.

Diaz-Uriarte R, Rueda OM: **ADaCGH: A parallelized web-based application and R package for the analysis of aCGH data**. *PLoS ONE* 2007g, **2**(8).

Eddy SR: **What is a hidden markov model**? *Nature Biotechnology* 2004, **22**(10):1315-1316.

Eichler EE: **Widening the spectrum of human genetic variation**. *Nature Genetics* 2006, **38**(1):9-11.

Engler DA, Mohapatra G, Louis DN, Betensky RA: **A pseudolikelihood approach for simultaneous analysis of array comparative genomic hybridizations**. *Biostatistics* 2006, **7**(3):399-421.

Feuk L, Carson AR, Scherer SW: **Structural variation in the human genome**. *Nature Reviews Genetics* 2006, **7**(2):85-97.

Ford CE, Jones KW, Polani PE, Dealmeida JC, Briggs JH: **A sex-chromosome anomaly in a case of gonadal dysgenesis (Turners Syndrome)**. *Lancet* 1959, **1**(4):711-713.

Freeman JL, Perry GH, Feuk L, Redon R, McCarroll SA, Altshuler DM, Aburatani H, Jones KW, Tyler-Smith C, Hurles ME: **Copy number variation: New insights in genome diversity**. *Genome Research* 2006, **16**(8):949-961.

Fridlyand J, Snijders AM, Pinkel D, Albertson DG, Jain AN: **Hidden markov models approach to the analysis of array CGH data**. *Journal of Multivariate Analysis* 2004, **90**(1):132-153.

Gelfand AE, Smith AFM: **Sampling-based approaches to calculating marginal densities**. *Journal of the American Statistical Association* 1990, **85**(410):398-409.

Gilks WR: **Markov chain Monte Carlo in practice**. London: Chapman & Hall; 1996.

Gonzalez E, Kulkarni H, Bolivar H, Mangano A, Sanchez R, Catano G, Nibbs R, Freedman B, Quinones M, Bamshad M: **The influence of CCL3L1 gene-containing segmental duplications on HIV-1/AIDS susceptibility**. *Science* 2005, **307**(5714):1434-1440.

Green PJ: **Reversible jump Markov chain Monte Carlo computation and bayesian model determination**. *Biometrika* 1995, **82**(4):711-732.

Guha S, Li Y, Neuberg D: **Bayesian hidden markov modeling of array CGH data**. *Journal of the American Statistical Association* 2008, **103**(482):485-497.

Hastings WK: **Monte Carlo sampling methods using markov chains and their applications**. *Biometrika* 1970, **57**(1):97-109.

Jacobs PA, Baikie AG, Brown WMC, Strong JA: **The somatic chromsomes in mongolism**. *Lancet* 1959, **1**(4):710-710.

Kang SJ, Gordon D, Finch SJ: **What SNP genotyping errors are most costly for genetic association studies**? *Genetic Epidemiology* 2004, **26**(2):132-141.

Kim W, Gordon D, Sebat J, Ye KQ, Finch SJ: **Computing power and sample size for case-control association studies with copy number polymorphism: application of mixture-based likelihood ratio test**. *PLoS ONE* 2008, **3**(10):e3475.

Lai WR, Johnson MD, Kucherlapati R, Park PJ: **Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data**. *Bioinformatics* 2005, **21**(19):3763-3770.

Mardis E: **The impact of next-generation sequencing technology on genetics**. *Trends in genetics: TIG* 2008, **24**(3):133-141.

Margulies M, Egholm M, Altman W, Attiya S, Bader J, Bemben L, Berka J, Braverman M, Chen Y-J, Chen Z: **Genome sequencing in microfabricated high-density picolitre reactors**. *Nature* 2005, **437**(7057):376-380.

Marioni JC, Thorne NP, Tavaré S: **BioHMM: a heterogeneous hidden markov model for segmenting array CGH data**. *Bioinformatics* 2006, **22**(9):1144-1146.

Matlab **Bioinformatics Toolbox Version 3.3**
[http://www.mathworks.com/matlabcentral]

Nguyen DQ, Webber C, Ponting CP: **Bias of selection on human copy-number variants**. *Plos Genetics* 2006, **2**(2):198-207.

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

Ordas J, Millan Y, Dios R, Reymundo C, Martin de las Mulas J: **Proto-oncogene HER-2 in normal, dysplastic and tumorous feline mammary glands: an immunohistochemical and chromogenic in situ hybridization study**. *BMC Cancer 2007*, **7**(1):179.

Picard F, Robin S, Lebarbier E, Daudin JJ: **A segmentation-clustering problem for the analysis of array CGH data**. *Applied Stochastic Models and Data Analysis 2005*, **6**(27):145-152.

Provost F, Kohavi R: **Guest editors' introduction: On applied research in machine learning**. *Machine Learning* 1998, **30**(2-3):127-132.

Rabiner LR: **A tutorial on hidden Markov models and selected applications in speech recognition**. *Proceedings of the IEEE* 1989, **77**(2):257-286.

Rabiner L, Juang B-H: **Fundamentals of speech recognition**: {Prentice Hall PTR}; 1993.

Redon R, Ishikawa S, Fitch K, Feuk L, Perry G, Andrews D, Fiegler H, Shapero M, Carson A, Chen W: **Global variation in copy number in the human genome**. *Nature* 2006, **444**(7118):444-454.

Rueda OM, Diaz-Uriarte R: **Flexible and accurate detection of genomic copy-number changes from aCGH**. *Plos Computational Biology* 2007, **3**(6):1115-1122.

Scott SL: **Bayesian methods for hidden Markov models: Recursive computing in the 21st century**. *Journal of the American Statistical Association* 2002, **97**(457):337-351.

Sebat J, Lakshmi B, Malhotra D, Troge J, Lese-Martin C, Walsh T, Yamrom B, Yoon S, Krasnitz A, Kendall J: **Strong association of de novo copy number mutations with autism**. *Science 2007*, **316**(5823):445-449.

Shah SP, Xuan X, Deleeuw RJ, Khojasteh M, Lam WL, Ng R, Murphy KP: **Integrating copy number polymorphisms into array CGH analysis using a robust HMM**. *Bioinformatics 2006*, **22**(14).

Shendure J, Porreca G, Reppas N, Lin X, McCutcheon J, Rosenbaum A, Wang M, Zhang K, Mitra R, Church G: **Accurate multiplex polony sequencing of an evolved bacterial genome**. *Science (New York, NY)* 2005, **309**(5741):1728-1732.

Shlien A, Malkin D: **Copy number variations and cancer susceptibility**. *Current Opinion Oncology* 2010, **22**(1):55-63.

Snijders AM, Nowak N, Segraves R, Blackwood S, Brown N, Conroy J, Hamilton G, Hindle AK, Huey B, Kimura K: **Assembly of microarrays for genome-wide measurement of DNA copy number**. *Nature Genetics* 2001, **29**(3):263-264.

Stjernqvist S, Ryden T, Skold M, Staaf J: **Continuous-index hidden markov modeling of array CGH copy number data**. *Bioinformatics* 2007, **23**(8):1006-1014.

Sudbery I, Stalker J, Simpson J, Keane T, Rust A, Hurles M, Walter K, Lynch D, Teboul L, Brown S: **Deep short-read sequencing of chromosome 17 from the mouse strains A/J and CAST/Ei identifies significant germline variation and candidate genes that regulate liver triglyceride levels**. *Genome Biology* 2009, **10**(10):R112.

Summitt RL: **Deletion of the short arm of chromosome 18**. *Cytogenetic and Genome Research* 1964, **3**(4):201-206.

Tanner M, Isola J, Wiklund T, Erikstein B, Kellokumpu-Lehtinen P, Malmstrom P, Wilking N, Nilsson J, Bergh J: **Topoisomerase IIalpha gene amplification predicts favorable treatment response to tailored and dose-escalated anthracycline-based adjuvant chemotherapy in HER-2/neu-amplified breast cancer: Scandinavian breast group trial 9401**. *Journal of Clinical Oncology* 2006, **24**(16):2428-2436.

van de Wiel MA, Kim KI, Vosse SJ, van Wieringen WN, Wilting SM, Ylstra B: **CGHcall: calling aberrations for array CGH tumor profiles**. *Bioinformatics* 2007, **23**(7):892-894.

Van Wieringen WN, Van De Wiel MA, Ylstra B: **Weighted clustering of called array CGH data**. *Biostatistics* 2008, **9**(3):484-500.

Venter JC, Adams MD, Sutton GG, Kerlavage AR, Smith HO, Hunkapiller M: **Shotgun sequencing of the human genome**. *Science* 1998, **280**(5369):1540-1542.

Viterbi AJ: **Error bounds for convolutional codes and an asymptotically optimal decoding algorithm**. *IEEE Transactions on Information Theory* 1967, **13**:260-269.

Wang Y, Wang S: **A novel stationary wavelet denoising algorithm for array based DNA Copy Number data**. *International Journal of Bioinformatics Research* 2007, **3**(2):206-222.

Willenbrock H, Fridlyand J: **A comparison study: applying segmentation to array CGH data for downstream analyses**. *Bioinformatics* 2005, **21**(22):4084-4091.

Xie C, Tammi M: **CNV-seq, a new method to detect copy number variation using high-throughput sequencing**. *BMC Bioinformatics* 2009, **10**(1):80.

Yoon S, Xuan Z, Makarov V, Ye K, Sebat J: **Sensitive and accurate detection of copy number variants using read depth of coverage**. *Genome Research* 2009, **19**(9):1586-1592.