A GENERAL GRAPHICAL FRAMEWORK FOR DETECTING COPY NUMBER VARIATIONS

Xiao-Lin Yin[1,2] and Jing Li [*1]

[1] *Electrical Engineering and Computer Science Department, Case Western Reserve University, Cleveland, OH 44106, USA*

[2] *KLAS of MOE, School of Mathematics and Statistics, Northeast Normal University, Changchun, Jilin 130024, China*

[*] *Email: jingli@case.edu*

Array comparative genomic hybridization (aCGH) allows identification of copy number alterations across genomes. The key computational challenge in analyzing copy number variations (CNVs) using aCGH data or other similar data generated by a variety of array technologies is the detection of segment boundaries of copy number changes and inference of the copy number state for each segment. We have developed a novel statistical model based on the framework of conditional random fields (CRFs) that can effectively combine data smoothing, segmentation and copy number state decoding into one unified framework. Our approach (termed CRF-CNV) provides great flexibilities in defining meaningful feature functions, therefore it can effectively integrate local spatial information of arbitrary sizes into the model. For model parameter estimations, we have adopted the conjugate gradient (CG) method for likelihood optimization and developed efficient forward/backward algorithms within the CG framework. The method is evaluated using real data with known copy numbers as well as simulated data with realistic assumptions, and compared with two popular publicly available programs. Experimental results have demonstrated that CRF-CNV outperforms a Bayesian Hidden Markov Model-based approach on both datasets in terms of copy number assignments. Comparing to a non-parametric approach, CRF-CNV has achieved much greater precision while maintaining the same level of recall on the real data, and their performance on the simulated data is comparable.

1. INTRODUCTION

Structure variations in DNA sequences such as inheritable copy number alterations have been reported to be associated with numerous diseases. It has also been observed that somatic chromosomal aberrations (*i.e.*, amplifications and deletions) in tumor samples have shown different clinical or pathological features in different cancer types or subtypes[3, 5, 11]. To gain more understanding of the role of inheritable copy number polymorphisms (CNPs) in determining disease phenotypes, systematic mapping and cataloging of CNPs are needed and are being carried out. Identification of somatic copy number aberrations in cancer samples may lead to the discovery of important oncogenes or tumor suppress genes. With remarkable capacity from current technologies in assessing copy number variants (CNVs), there is a great wave of interests recently from the research community to investigate inheritable as well as somatic CNVs[1, 3–5, 9, 11, 12, 17]. Broadly speaking, there are essentially three technological platforms for copy number variation detections: array-based technology (including array comparative genomic hybridization (aCGH), as well as many other variants such as oligonucleotide array or bacterial artificial chromosome array), SNP genotyping technology[1, 11] and next-generation sequencing technology[2]. Array-based technology measures DNA copy number changes from a disease sample relative to a normal sample, represented by the log_2 ratio of corresponding fluorescence intensities of each clone. The log_2 ratios are expected to be proportional to copy numbers, though significant noise can be introduced from various sources. Array-based technologies are primarily for large segments of duplications/deletions, though different experimental platforms and designs using clones with different sizes, may give very different resolutions and coverages[11]. To identify small CNVs at a finer resolution, researchers have been using raw intensity values of SNPs that interrogate CNVs. This technique has gained great popularity with the availability of SNP chips from major vendors. More recently, new algorithms have been proposed to identify CNVs from new genotyping platforms by integrating information from both SNP probes and CNV probes[13]. Yet an-

[*]Corresponding author.

other alternative is on the horizon, *i.e.*, one can use massively parallel sequencing techniques to identify CNVs with even finer resolutions. A very few preliminary studies[2] have shown it has better power to localize breakpoints.

Different platforms have different challenges. Naturally, one should use different approaches for these platforms by taking advantage of special properties from different datasets. Not surprisingly, various algorithms have been proposed for different data in recently years. On the other hand, the primary goal of all such studies is to identify and localize the copy number changes, therefore, the essential computational task for data from different platforms is the same: to segment genomes into discrete regions of CNVs based on the measurements of probe intensity values and/or ratios, or number of sequence reads. One important commonality in data from different platforms is the spatial correlation among clones/probes/sequences. Many existing approaches have taken advantage of such a property by utilizing the same methodology, Hidden Markov Models (HMMs), which can conveniently model spatial dependence using a chain structure. Results have shown initial success[1, 4, 12, 17] of HMMs. However, there is an inherited limitation for all these HMMs, *i.e.*, they are all first order HMMs and cannot take into consideration long range dependence. We propose to develop and apply a novel undirected graphical model based on Conditional Random Fields (CRFs)[16] for the segmentation of CNVs. It has been shown that CRFs consistently outperform HMMs in a variety of applications, mainly because CRFs can potentially integrate all information from data[16]. This property makes CRFs particularly appealing to model CNV data since one can define feature functions using data from a region rather than a single or two data points for emissions and transitions, respectively, in HMMs.

Our major analytical contributions include the construction of the CRF model, the definition of effective feature functions using robust statistics, and the development of efficient computation algorithms for parameter estimations. As an illustration of our proposed model, we have applied our approach on real and simulated data based on array technology, and compared its performance with two popular segmentation algorithms. Experimental results have demonstrated that CRF-CNV outperforms a Bayesian Hidden Markov Model-based approach on both datasets in terms of copy number assignments, but with little sacrifice of accuracy in breakpoint identification due to smoothing. Comparing to a non-parametric approach, CRF-CNV has achieved much greater precision while maintaining the same level of recall on the real data. On the simulated data, CRF-CNV has obtained better accuracy in identifying breakpoints with comparable performance in copy number assignments. The reminder of this article is organized as follows. In Section 2, we give a brief overview of aCGH data and existing approaches for detecting CNVs from aCGH data. We also briefly mention the differences between HMMs and CRFs. Details about model developments and implementations are provided in Section 3. Our experimental results on two datasets and comparisons with other two programs are presented in Section 4. We conclude the paper with a few discussions in Section 5.

2. PRELIMINARY

2.1. aCGH Data and Analysis

Though theoretically, our approach can be applied to data from different experimental platforms. We focus primarily on aCGH data in this analysis. Mathematically, aCGH data usually consist of an array of log_2 intensity ratios for a set of clones, as well as the physical position information of each clone along a genome. Figure 1 plots the normalized log_2 ratio of one cell line (GM04435) analyzed by Snijders *et al.*[14]. Each data point represents one clone and the y-axis represents normalized log_2 intensity ratio. The primary goal in CNV detection based on aCGH is to segment a genome into discrete regions that share the same mean log_2 ratio pattern (*i.e.*, have the same copy numbers). Ideally, the log_2 ratio of a clone should be 0 if the cancer sample/cell line has a normal number (*i.e.*, 2) copies of DNA, and the value should be around 0.585 (or -1) if it has one copy of gain (or loss). However, as shown in Figure 1, aCGH data can be quite noisy with vague boundaries between different segments. It may also have complex local spatial dependence structure. These properties

make the segmentation problem intrinsically hard. Approaches using a global threshold generally do not work in practice.

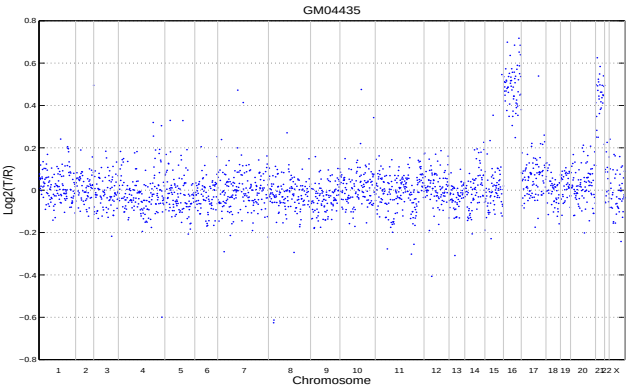


Fig. 1. Array CGH profile of a Corriel cell line (GM04435). The borders between chromosomes are indicated by grey vertical bars.

2.2. Existing Algorithms

In general, a number of steps are needed to detect copy number changes from aCGH data. First, raw log_2 ratio data usually needs some preprocessing, including normalization and smoothing. Normalization is an absolute necessary step to alleviate systemic errors due to experimental factors. Usually the input data is normalized by making the median or mean log_2 ratio of a selected median set from normal copy number regions to be zero. Smoothing is used to reduce noises that are due to random errors or abrupt changes. Smoothing methods generally filter the data using a sliding window, attempting to fit a curve to the data while handling abrupt changes and reducing random errors. A number of smoothing methods have been proposed, such as sliding window smoothing, quantile smoothing, wavelet smoothing, *etc.*

The second step in analyzing aCGH data is referred to as *segmentation* and it aims to identify contiguous sets of clones (segments) that share the same mean log_2 ratio. Broadly, there are two related estimation problems. One is to infer the number and statistical significance of the alterations; the other is to locate their boundaries accurately. A few different algorithms have been proposed to solve these two estimation problems. Olshen *et al.*[10] have proposed a non-parametric approach based on the recursive circular binary segmentation (CBS) algorithm. Hupe *et al.*[7] have proposed an approach called GLAD, which is based on a median absolute deviation model to separate outliers from their surrounding segments. Willenbrock and Fridlyand[18] have compared the performance of CBS (implemented in DNACopy) and GLAD using a realistic simulation model, and they have concluded that CBS in general is better than GLAD. We will also adopt their simulation model in our experiment study. After obtaining the segmentation outcomes, a postprocessing step is needed to combine segmentations with similar mean levels and to classify them as single-copy gain, single-copy loss, normal, multiple gains, *etc.* Methods such as GLADMerge[7] and MergeLevels[18] can take the segmentation results and label them accordingly.

As noted by Willenbrock and Fridlyand[18], it is more desirable to perform segmentation and classification simultaneously. An easy way to merge these two steps is to use a linear chain HMM. The underlying hidden states are the real copy numbers. Given a state, the log_2 ratio can be modeled using a Gaussian distribution. The transition from one state to another state reveals the likelihood of copy number changes between adjacent clones. Given observed data, standard algorithms (forward/backward, Baum-Welch and Viterbi) can be used to estimate parameters and to decode hidden states. A few variants of HMMs have been proposed for aCGH data in recent years[6, 12]. Guha *et al.*[6] have proposed a Bayesian HMM which can impose biological meaningful priors on the parameters. Shah *et al.*[12] have extended this Bayesian HMM by adding robustness to outliers and location-specific priors, which can be used to model inheritable copy number polymorphisms. Notice that all these models are first-order HMMs which cannot capture long range dependence. Intuitively, it makes sense to consider high-order HMMs to capture informative local correlation, which is an important property observed from aCGH data. However, considering higher orders will make HMMs more complex and computationally intensive.

2.3. Conditional Random Fields

To overcome the limitations of HMMs, we propose a new model based on the theory of Conditional Random Fields (CRFs). CRFs are undirected graphical models designed for calculating the conditional distribution of output random variables Y given input variables X[16]. It has been extensively applied to language processing, computer vision, and bioinformatics with remarkable performance comparing with directed graphical models including HMMs. The key difference between CRFs and HMMs is that one can define meaningful *feature functions* that can effectively capture local spatial dependence among observations. As illustrated in Figure 2, although we also use a chain structure in our CRF model for CNV detection, our feature functions to be defined can use observed data from a region. Therefore it can capture abundant local spatial dependence. In addition, by using a CRF, we can effectively combine smoothing, segmentation and classification into one unified framework.

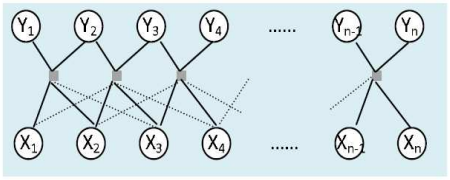


Fig. 2. A linear chain conditional random field model for array CGH data.

3. METHODS

3.1. Linear-Chain CRF Model for aCGH Data

Our model is based on the linear-chain CRF Model in Figure 2. Let $X = (X_1, \ldots, X_n)$ denote the normalized log_2 ratio intensities along one chromosome for an individual, where X_i is the log_2 ratio for clone i. One can assume that these n clones are sequentially positioned on a chromosome. Let $Y = (Y_1, \ldots, Y_n)$ denote the corresponding hidden copy number state, where $Y_i \in \{1, .., s\}$ and s is the total number of copy number states. These states usually indicate deletion, single-copy loss, neutral, single-copy gain, two-copy gain or multiple-copy gain. The exact number of states and their meaning need to be specified based on specific input data. The conditional probability of Y given observed log_2 ratio X based on our linear-chain CRF structure can be defined as

$$P(Y|X) = \frac{1}{Z_\theta(X)} exp\{\sum_{i=1}^{n}\sum_{j=1}^{s}[\lambda_j f_j(Y_i, \tilde{X}_i(u)) + \mu_j g_j(Y_i, \tilde{X}_i(u))] + \sum_{j=1}^{s}\omega_j l_j(Y_1, \tilde{X}_1(u)) + \sum_{i=1}^{n-1}\sum_{j=1}^{s}\sum_{k=1}^{s}\nu_{jk} h_{jk}(Y_i, Y_{i+1}, \tilde{X}_{i,i+1}(u))\}, \quad (1)$$

where the partition function

$$Z_\theta(X) = \sum_{Y} exp\{\sum_{i=1}^{n}\sum_{j=1}^{s}[\lambda_j f_j(Y_i, \tilde{X}_i(u)) + \mu_j g_j(Y_i, \tilde{X}_i(u))] + \sum_{j=1}^{s}\omega_j l_j(Y_1, \tilde{X}_1(u)) + \sum_{i=1}^{n-1}\sum_{j=1}^{s}\sum_{k=1}^{s}\nu_{jk} h_{jk}(Y_i, Y_{i+1}, \tilde{X}_{i,i+1}(u))\}.$$

Here $\theta = \{\lambda_j, \mu_j, \omega_j, \nu_{jk}\}$ are parameters that need to be estimated. Functions f_j, g_j, l_j and h_{jk} are feature functions that need to be defined. $\tilde{X}_i(u)$ is defined as a neighbor set of X_i around clone i, *i.e.*, $\tilde{X}_i(u) = \{X_{i-u}, ..., X_{i-1}, X_i, X_{i+1}, ..., X_{i+u}\}$, where u is a hyper-parameter to define the dependence length. Similarly, we define $\tilde{X}_{i,i+1}(u) = \{X_{i-u}, ..., X_i, X_{i+1}, ..., X_{i+1+u}\}$, $\tilde{X}^-_{i,i+1}(u) = \{X_{i-u}, ..., X_{i-1}, X_i\}$ and $\tilde{X}^+_{i,i+1}(u) = X_{i+1}, X_{i+2}, ..., X_{i+1+u}\}$. We will define our feature functions by integrating all information from these neighborhood sets. The dependence length u plays a similar role like the width of a sliding window in smoothing methods. For notational simplification, we drop the parameter u in our subsequent discussions and write $\tilde{X}_i(u)$ as $\tilde{X}_i$ and *etc.*

3.2. Feature Functions

One important step to build our model is to define meaningful feature functions that can capture critical information from input data. Essentially, we define two types of feature functions, analogous to the emission and transition probabilities in HMMs. However,

our feature functions can be of any form, therefore our model can provide much more flexibilities and be able to capture long range dependence. The emission feature functions $f_j(Y_i, \tilde{X}_i)$ and $g_j(Y_i, \tilde{X}_i)$ are defined as follows:

$$f_j(Y_i, \tilde{X}_i) = \begin{cases} med\ \tilde{X}_i \text{ if } Y_i = j \\ 0 \qquad\quad \text{otherwise,} \end{cases}$$

$$g_j(Y_i, \tilde{X}_i) = \begin{cases} (med\ \tilde{X}_i)^2 \text{ if } Y_i = j \\ 0 \qquad\quad \text{otherwise,} \end{cases}$$

where $med\ \tilde{X}_i$ is defined as the median value of set $\tilde{X}_i$. Our emission features sever two purposes. First it is used as a median filter that will automatically smooth the input data. More importantly, the feature functions based on the first-order and second-order median statistics are robust sufficient statistics one can derive from a normal distribution, which resemble the emission pattern of log_2 ratio intensities for a given hidden copy number state.

The transition feature function $h_{jk}(Y_i, Y_{i+1}, \tilde{X}_{i,i+1})$ and the initial feature function $l_j(Y_1, \tilde{X}_1)$ are defined as follows(h is on the next page):

$$l_j(Y_1, \tilde{X}_1) = \begin{cases} \frac{(a_{j+1}-a_j)/2}{(a_{j+1}-a_j)/2+med\ \tilde{X}_1-a_j} \text{ if } Y_1 = j, \\ \qquad\qquad\qquad\qquad med\ \tilde{X}_1 \geq a_j \\ \frac{(a_j-a_{j-1})/2}{(a_j-a_{j-1})/2+a_j-med\ \tilde{X}_1} \text{ if } Y_1 = j, \\ \qquad\qquad\qquad\qquad med\ \tilde{X}_1 < a_j \\ 0 \qquad\qquad\qquad\qquad \text{otherwise.} \end{cases}$$

Here a_j denotes the mean log_2 ratio for clones with copy number state j ($j = 1, \ldots, s$). While a_0 and a_{s+1} denote the greatest lower bound of log_2 ratio for clones with copy number state 1 and the least upper bound of log_2 ratio for clones with copy number state s, respectively. Without loss of generality, we assume $a_0 < a_1 < \ldots < a_{s+1}$. We define the initial feature function $l_j(Y_1, \tilde{X}_1)$ in a way such that data from the clone set $\tilde{X}_1$ will only provide information to its own labelled state. Furthermore, when $Y_1 = j$, the closer the $med\ \tilde{X}_1$ to a_j, the higher value for $l_j(Y_1, \tilde{X}_1)$, more information data will provide and more contribution to parameter ω_j. It will achieve the highest value of 1 when $med\ \tilde{X}_1 = a_j$. The transition feature function $h_{jk}(Y_i, Y_{i+1}, \tilde{X}_{i,i+1})$ is similarly defined using the clone set $\tilde{X}_{i,i+1}$. When $Y_i = j$ and $Y_{i+1} = k$, the closer the $med\ \tilde{X}^-_{i,i+1}$ to a_j and the $med\ \tilde{X}^+_{i,i+1}$ to a_k, the higher value for $h_{jk}(Y_i, Y_{i+1}, \tilde{X}_{i,i+1})$, and more information the data will contribute to ν_{jk}. Clearly, both types of our feature functions can capture the local spatial dependence over a set of adjacent clones thus potentially provide more robust inference about hidden copy number states.

3.3. Parameter Estimation

Unlike the standard algorithms for HMM training, there are significant computational challenges to efficiently and accurately estimate parameters for CRFs. Implementation of the training algorithms for our proposed CRF model requires sophisticated statistical and numerical algorithms. To our best knowledge, no existing implementations can be used to solve our problem. We propose the following algorithm for the parameter estimation.

In general, given a set of training data $\mathcal{D} = \{(X^{(d)}, Y^{(d)}), d = 1, \ldots, D\}$, to estimate parameter θ in model (1), one needs to maximize a penalized conditional log likelihood which is defined as follows

$$\begin{aligned} L_\theta &= \sum_{d=1}^{D} logP(Y^{(d)}|X^{(d)}) - \frac{\| \theta \|^2}{2\sigma^2} \\ &= \sum_{j=1}^{s} \lambda_j \sum_{d=1}^{D}\sum_{i=1}^{n} f_j(Y_i^{(d)}, \tilde{X}_i^{(d)}) + \sum_{j=1}^{s} \mu_j \sum_{d=1}^{D}\sum_{i=1}^{n} \\ &\quad g_j(Y_i^{(d)}, \tilde{X}_i^{(d)}) + \sum_{j=1}^{s} \omega_j \sum_{d=1}^{D} l_j(Y_1^{(d)}, \tilde{X}_1^{(d)}) + \\ &\quad \sum_{j=1}^{s}\sum_{k=1}^{s} \nu_{jk} \sum_{d=1}^{D}\sum_{i=1}^{n-1} h_{jk}(Y_i^{(d)}, Y_{i+1}^{(d)}, \tilde{X}_{i,i+1}^{(d)}) - \\ &\quad \sum_{d=1}^{D} logZ_\theta(X^{(d)}) - \frac{\| \theta \|^2}{2\sigma^2}. \end{aligned} \tag{2}$$

Here $\| \theta \|$ is the L^2 norm of θ. The penalization term $\| \theta \|^2 / 2\sigma^2$ is added for regularization purpose. Before one can solve the optimization problem, one has to first specify an additional set of hyper-parameters that include the dependence length u, the mean log_2 ratios $\{a_j, j = 0, \ldots, s+1\}$ and the penalization coefficient σ^2. The set of $\{a_j\}$ can be directly estimated given the training data set $\mathcal{D}$, *i.e.*, the maximum likelihood estimate of a_j is just the mean value log_2 ratios of all clones with copy number state j in $\mathcal{D}$ for $j = 1, \ldots, s$. While a_0 and a_{s+1} can be imputed using the minimum log_2 ratio of all clones with copy number state 1, and the maximum value from all

$$h_{jk}(Y_i, Y_{i+1}, \tilde{X}_{i,i+1}) =$$

$$\begin{cases} \frac{(a_{j+1}-a_j)/2+(a_{k+1}-a_k)/2}{(a_{j+1}-a_j)/2+(a_{k+1}-a_k)/2+med\ \tilde{X}^-_{i,i+1}-a_j+med\ \tilde{X}^+_{i,i+1}-a_k} & \text{if } Y_i = j,\ med\ \tilde{X}^-_{i,i+1} \geq a_j, \\ & Y_{i+1} = k,\ med\ \tilde{X}^+_{i,i+1} \geq a_k \\ \frac{(a_{j+1}-a_j)/2+(a_k-a_{k-1})/2}{(a_{j+1}-a_j)/2+(a_k-a_{k-1})/2+med\ \tilde{X}^-_{i,i+1}-a_j+a_k-med\ \tilde{X}^+_{i,i+1}} & \text{if } Y_i = j,\ med\ \tilde{X}^-_{i,i+1} \geq a_j, \\ & Y_{i+1} = k,\ med\ \tilde{X}^+_{i,i+1} < a_k \\ \frac{(a_j-a_{j-1})/2+(a_{k+1}-a_k)/2}{(a_j-a_{j-1})/2+(a_{k+1}-a_k)/2+a_j-med\ \tilde{X}^-_{i,i+1}+med\ \tilde{X}^+_{i,i+1}-a_k} & \text{if } Y_i = j,\ med\ \tilde{X}^-_{i,i+1} < a_j, \\ & Y_{i+1} = k,\ med\ \tilde{X}^+_{i,i+1} \geq a_k \\ \frac{(a_j-a_{j-1})/2+(a_k-a_{k-1})/2}{(a_j-a_{j-1})/2+(a_k-a_{k-1})/2+a_j-med\ \tilde{X}^-_{i,i+1}+a_k-med\ \tilde{X}^+_{i,i+1}} & \text{if } Y_i = j,\ med\ \tilde{X}^-_{i,i+1} < a_j, \\ & Y_{i+1} = k,\ med\ \tilde{X}^+_{i,i+1} < a_k \\ 0 & \text{otherwise,} \end{cases}$$

clones with copy number state s, respectively. For the dependent length u and the penalization coefficient σ^2, we rely on a grid search approach through cross-validation. More specifically, the original training set $\mathcal{D}$ will first be partitioned into two sets $\mathcal{D}_1$ and $\mathcal{D}_2$. We call $\mathcal{D}_1$ the new training set and $\mathcal{D}_2$ the validation set. For a given range of (discrete) parameter values of u and σ^2, we train the model on $\mathcal{D}_1$ and get estimates of θ for each fixed pair of (u_0, σ_0^2). The exact procedure to estimate θ given (u_0, σ_0^2) will be discussed shortly. We then apply the trained model with estimated parameters on the validation set $\mathcal{D}_2$ and record the prediction errors under the current model. The model with the smallest prediction error as well as their associated parameters (u, σ^2, θ) will be chosen as the final model. The prediction error is defined as the mean absolute error (MAE) for all samples in the validation set $\mathcal{D}_2$. The absolute error for a clone i is defined as $\mid Y_i - \hat{Y}_i \mid$, where Y_i is the known copy number and $\hat{Y}_i$ is the predicted copy number for clone i. This measure not only captures whether a prediction is exactly the same as the real copy number, but also reflects how close these two numbers are.

For a given set of hyper-parameters $\{a_j\}$, u and σ^2, the optimization of L_θ in equation (2) can be solved using gradient-based numerical optimization methods[15]. We choose the nonlinear Conjugate Gradient (CG) method in our implementation, which only requires to compute the first derivatives of L_θ. The partition function $Z_\theta(X)$ in the log likelihood and the marginal distributions in gradient functions can be computed using forward-backward algorithms. Due to page limitation, technical details of the CG method and the efficient computation of the derivatives of L_θ will be provided in the final journal version.

For graphical model based approaches such as HMMs, many researchers group both individuals and chromosomes in the analysis of aCGH data, which can dramatically reduce the number of parameters needed without sacrificing much on inference accuracy. We also take a similar approach. This is reflected by our homogeneous CRF structure.

3.4. Evaluation Methods

We have implemented the above proposed approach as a Matlab package termed CRF-CNV and evaluated its performance using a public available real data set with known copy numbers[14] and a synthetic data set from Willenbrock and Fridlyand[18]. Notice that many clones have normal (2) copies of DNAs, therefore the number of correctly predicted state labels is not a good measure of performance of an algorithm. Instead, we compare the performance of CRF-CNV with two popular programs in terms of the number of predicted segments and the accuracy of segment boundaries, referred to as breakpoints. To summarize the performance of an algorithm over multiple chromosomes and individuals, we use a single value called $F-$measure, which is a combination of *precision* and *recall*. Recall that given the true copy number state labels and predicted labels, precision (P) is defined as $\frac{ntp}{np}$ and recall (R) is defined as $\frac{ntp}{nt}$, where ntp is the number of true positive (correctly predicted breakpoints), np is the number of predicted breakpoints, and nt is

the number of true breakpoints. $F-$measure is defined as $F = 2PR/(P+R)$, which intends to find a balance between precision and recall. The two programs we chose are CBS[10] and CNA-HMMer[12], both of which have been implemented as Matlab tools. As mentioned earlier, CBS is one of the most popular segmentation algorithms and different groups have shown it in general performs better than many other algorithms. CNA-HMMer is chosen because we want to compare the performance of our CRF model with HMMs, and CNA-HMMer is an implementation of Bayesian HMM model with high accuracy[12].

4. EXPERIMENTAL RESULTS

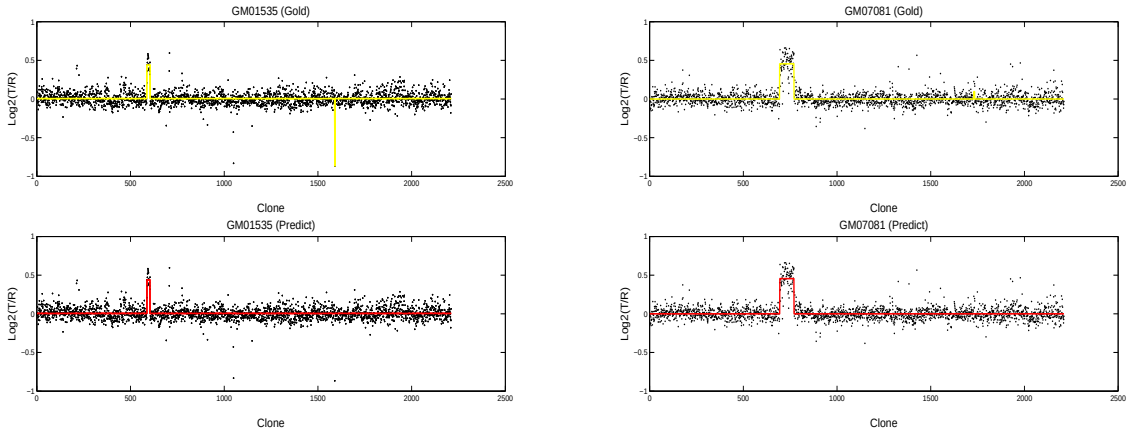
4.1. A Real Example

The Coriell data is regarded as a well-known "gold standard" data set which was originally analyzed by Snijders et al.[14]. The data is publicly available and has been widely used in testing new algorithms and in comparing different algorithms. The CBS algorithm has been applied on this dataset in the original paper. We redo the analysis using the Matlab code to obtain a complete picture. The Coriell data consists of 15 cell lines, named GM03563, GM00143, ..., GM01524. We simply use number $1, 2, \ldots, 15$ to represent these cell lines. For this particular dataset, there are only three states ($s = 3$), *i.e.*, loss, neutral and gain. Notice that unlike CBS, CRF-CNV requires training data to obtain parameters. It is unfair to directly compare the prediction results of CRF-CNV on training data with results from CBS. We take a simple approach which divides the 15 samples into three groups. Each group consists of 5 samples. In the first run, we use group 1 as training data and group 2 as validation data to obtain model parameters (as discussed in subsection 3.3). We then use the model to predict data in group 3 (testing data), and record the prediction results. In the second and third run, we alternate the roles of groups 1-3 and obtain prediction results of samples in group 1 and group 2, respectively. Finally we summarize our results over all 15 samples. For example, for the first run, we first obtain $\{a_j, j = 0, \ldots, 4\}$ directly based on samples in group 1. The estimates of $\{a_j\}$ is (-1.348, -0.682, -0.001, 0.497, 0.810). To search the penalization coefficient σ^2 and the dependent length u, we define the search space as $A \times B = \{0, 1, 2, \ldots, 30\} \times \{0, 1, \ldots, 5\}$. For each data point $(m, u_0) \in A \times B$, we let $\sigma^2 = 400 \times 0.8^m$ and $u = u_0$. Essentially to search σ^2 in a broad range, we use a geometric decay. The upper bounder on u is set to be 5 because for aCGH data such as the Coriell dataset, each clone can cover a quite long range of DNA. The optimal σ^2 and u will be chosen by minimizing the prediction errors on samples in Group 2 (the validation set). Our results indicate that the model with $u = 1$ and $m = 21$ achieves the lowest prediction error. Notice that $u = 1$ implies feature functions are defined based on a window size of 3. The values of θs can be estimated simultaneously. We then apply Viterbi's algorithm to find the most possible hidden copy number states for samples in Group 3, as well as the number and boundaries of segments. Run 2 and run 3 will obtain results on group 1 and group 2. For the CNA-HMMer, one can either use its default priors, or use training data to obtain informative priors. We have tested the performance of CNA-HMMer both with and without informative priors.

Table 1 shows the segment numbers of each sample from the Gold Standard, and from the predicted outcomes of the three algorithms CRF-CNV, CBS and CNA-HMMer. The segment number detected by CRF-CNV is exactly the same as the Gold Standard for almost all samples (except for sample 9 and 10). Further examination of samples 9 and 10 (see Figure 3) reveals that the segment that we missed in sample 9 only has one clone, which has been smoothed out by our algorithm. The segment missed in sample 10 is also very short and the signal is very weak. Our results have shown that CBS has generated many more segments comparing to the ground truth, which is consistent with the results in the original paper. The overall number of segments reported by CNA-HMMer with default priors is even greater than the total number from CBS. On the other hand, once we have used training data to properly assign informative priors for CNA-HMMer, it almost returns the same number of segments as CRF-CNV. The only exception is that CNA-HMMer missed one breakpoint in sample one. This illustrates that by using correctly labeled training data, both CRF-CNV and CNA-HMMer can effectively eliminate all false

Table 1. Comparison of segment numbers returned by three algorithms.

method ╲ sample	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	sum
Gold	5	3	5	3	5	3	5	5	5	5	3	5	2	3	3	60
CRF-CNV	5	3	5	3	5	3	5	5	3	3	3	5	2	3	3	56
CBS	17	42	7	6	9	5	5	6	5	5	13	7	17	3	9	156
CNA-HMMer(default)	9	83	9	7	11	3	7	5	11	11	21	5	16	10	16	224
CNA-HMMer (trained)	3	3	5	3	5	3	5	5	3	3	3	5	2	3	3	54



(a) (b)

Fig. 3. Predicted breakpoints by CRF-CNV (bottom) vs. true breakpoints (up) on two cell lines GM01535 (a) and GM07081 (b).

positives in this dataset. For the subsequent experiments, we only report the results of CNA-HMMer with proper training.

Table 2. Comparison of F measure with different match extent values for three algorithms.

method ╲ match extent	CRF-CNV	CNA-HMMer	CBS
0	0.638	0.877	0.333
1	0.914	0.947	0.500
2	0.948	0.947	0.519
3	0.967	0.947	0.519
4	0.967	0.947	0.519

As a comparison measure, the number of segments is a very rough index because it does not contain information about breakpoints. To further examine how accurate the predicted breakpoints by each approach, we pool all the breakpoints from all the samples and use the F measure defined earlier to compare the performance of the three algorithms. Notice that even though exact matches are possible, shifting by a few clones around boundaries is also likely given noisy input data. Therefore we use a match extent index D to allow some flexibility in defining matches of predicted breakpoints to those given by the gold standard. Table 2 shows F measures given different match extent values for the three methods. Clearly, CBS has the worst performance regardless the match extent values. This partially reflects that it has many false positives. The results from CNA-HMMer are very accurate when no match extent is allowed and then it shows modest increase when we increase the value of D from 0 to 1. The results of CRF-CNV lie in between when the match index $D = 0$. However, the performance of CRF-CNV is greatly enhanced when $D = 1$ and finally it outperforms CNA-HMMer when $D \geq 2$. The primary reason that CRF-CNV has shifted one or a few positions for many breakpoints is because of the automatic median smoothing step. In contrast, CNA-HMMer directly models outliers using prior distributions.

4.2. Simulated Data

Though results on the real data have shown that CRF-CNV has a better performance than CBS and CNA-HMMer, the experiment is limited because the sample size is very small. To further evaluate the performance of CRF-CNV, we test the three algorithms using a simulated dataset obtained from Willenbrock and Fridlyand[18]. The dataset consists of 500 samples each with 20 chromosomes. Each chromosome contains 100 clones. Each clone belongs to one of six possible copy number states. The authors generated these samples by sampling segments from a primary breast tumor data set of 145 samples and used several mechanisms (*e.g.*, the fraction of cancer cells in a sample, the variation of intensity values given a copy number state) to control the noise level. By using simulated data from the literature, we can obtain an un-biased picture about CRF-CNV's performance. The original paper also compared three algorithms and concluded that CBS has the best performance.

To train CRF-CNV, we divide the 500 samples into three groups as usual. This time, the training set Group 1 contains sample 1-50, the validation set Group 2 contains sample 51-100 and the test set Group 3 contains sample 101-500. We use the same grid search approach as discussed earlier to obtain hyper-parameters $\{a_j\}$, u and σ^2. For each fixed set of hyper-parameters, we use the conjugate gradient method to obtain parameter θ. Finally, we use Viterbi's algorithm to decode the most possible hidden copy number state labels for samples in Group 3 and compare the results with the other two algorithms. In addition, we also compare the predictions by CRF-CNV on group 2 and group 3 to see that on new testing data, how much deterioration our model might incur based on sub-optimal parameters inferred from small number of samples. Results from CBS and CNA-HMMer are also presented separately for these two groups for easy comparison. We also use Group 1 as training data to assign proper priors for CNA-HMMer.

Table 3. Comparison of number of segments predicted by three different approaches.

method ╲ data	Group 2	Group 3
Gold	997	8299
CRF-CNV	966	8868
CNA-HMMer	784	6692
CBS	867	7430

Table 3 shows the total number of segments in Group 2 and Group 3 predicted by CRF-CNV, CBS and CNA-HMMer, and in comparison with the known segment number. Interestingly, on this simulated data, both CBS and CNA-HMMer have predicted smaller number of segments. CRF-CNV has predicted smaller number of segments on group 2 and greater number of segments in group 3. However, the number of segments does not provide a whole picture. We therefore examine the accuracy of boundary prediction by each method using the F measure for both Group 2 and Group 3. Table 4 shows the F measures for different methods, different groups and different match extents. As expected, the F measure increases as D increases from 0 to 4 for all methods and for both data groups. It is also not surprising to see that the results of CBS and CNA-HMMer on Group 2 and Group 3 are consistent. Interestingly, the performance of CRF-CNV on Group 3 is also very close to its own performance on Group 2. This property is desirable because it illustrates the robustness of CRF-CNV. The performance on new testing data is almost the same as the performance on validation data, which is used to select optimal hyper-parameters. This observation alleviates the need of training samples by our approach and makes it more practical. Notice that the sizes of training data and validation data are also very small. One can expect that with small number of training data, our approach can be used to reliably predict new data generated under the same experimental conditions. In terms of the performance of the three approaches, CNA-HMMer is more accurate then CRF-CNV, and CBS is the worst for the case of exact match. However, when we relax the matching criteria by increasing the value of D, both CBS and CRF-CNV achieves better performance than CNA-HMMer. The results of CNA-HMMer and CRF-CNV is consistent with those from the real data. CBS has much better per-

Table 4. Comparison of F measure of different methods with different match extent.

method \ match extent	0	1	2	3	4
CRF-CNV(Group2)	0.590	0.792	0.875	0.900	0.906
CNA-HMMer(Group2)	0.702	0.801	0.832	0.852	0.855
CBS(Group2)	0.436	0.850	0.885	0.900	0.909
CRF-CNV(Group3)	0.568	0.786	0.864	0.889	0.896
CNA-HMMer(Group3)	0.697	0.805	0.840	0.858	0.869
CBS(Group3)	0.436	0.847	0.893	0.911	0.918

formance comparing to those from the real data. But this might be attributed to the simulation process because CBS was used to segment the 145 samples from the primary breast tumor data set[18].

5. CONCLUSION AND DISCUSSIONS

The problem of detecting copy number variations has drawn much attention in recent years and many approaches have been proposed to solve the problem. Among these computational developments, CBS has gain much popularity and it has been shown that it generally performs better than other algorithms on simulated data[18]. However, as shown in the original paper (as well as re-discovered by our experiments), CBS has reported many more false positives on copy number changes in the standard Coriell data set identified by spectral karyotyping[14]. Another commonly used technique for segmentation is HMMs. HMM approaches have the advantage of performing parameter (*i.e.*, means and variances) estimation and copy number decoding within one framework and its performance expects to be improving with more observations. Furthermore, Lai et al.[8] have shown that HMMs performed the best for small aberrations given a sufficient signal/noise ratio. However, almost all HMMs for aCGH are first order Markov models thus cannot incorporate long region spatial correlations within data.

We have presented a novel computational model based on the theory of conditional random fields. We have also developed effective forward/backward algorithms within the conjugate gradient method for efficient computation of model parameters. We evaluated our approach using real data as well as simulated data, and results have shown our approach performed better than a Bayesian HMM on both datasets when a small shift is allowed while mapping breakpoints. A further discussion on the relationship between our proposed CRF model and a HMM can be found in APPENDIX. Comparing with CBS, our approach has much less false positives on the real data set. On the simulated data set, the performance of our approach is comparable to CBS, which has been shown the best among three popular segmentation approaches.

Like any other CRFs, in order to train our model, one has to rely on some training data. To be practically useful, Bayesian HMMs such as CNA-HMMer also need training data for proper assignments of informative priors. We argue that the problem is not that serious as it appears to be, primarily for two reasons. First, as illustrated in our experiments, our algorithm is indeed very robust and performs consistently even one may not find the optimal estimates of model parameters. For example, we used a simplified procedure in the analysis of the simulated dataset by randomly picking one subset for training. Theoretically, parameters estimated from such a procedure might heavily depend on this particular subset and might not be necessarily globally optimal. However, results in Table 4 have shown that the performance on new testing data is almost the same as the results in the verification data, which has been used to tune the parameters. Furthermore, the training size required by our algorithm is very small, as illustrated by both the real and the simulated data.

In terms of computation costs, CNV-CRF has two separate portions: time for training and time for prediction. The training requires intensive computations in optimizing the log-likelihood and in determining the hyper-parameters. In addition, one can also perform k-fold cross-validations, which will require much more computational time. On the contrary, once the parameters have been estimated, the prediction phase is rather efficient. Fortunately, the training phase of our algorithm only requires small

number of samples, which makes the algorithm still practically useful.

ACKNOWLEDGEMENTS

This work is supported in part by NIH/NLM (grant LM008991), NIH/NCRR (grant RR03655), NSF (grant CRI0551603) and a start-up fund from Case Western Reserve University. We appreciate Matthew Hayes for helpful discussions.

APPENDIX

Relationship of CRFs and HMMs

A special case of our linear-chain CRF model defined in subsection 3.1 corresponds to a familiar HMM. For example, let $\lambda_j = c_j/d_j^2$, $\mu_j = -1/(2d_j^2)$, $\omega_j = \log P(Y_1 = j)$, $\nu_{jk} = \log P(Y_{i+1} = k|Y_i = j)$, $f_j(Y_i, \tilde{X}_i(u)) = I_{\{Y_i=j\}} med\ \tilde{X}_i$, $g_j(Y_i, \tilde{X}_i(u)) = (med\ \tilde{X}_i)^2$, $l_j(Y_1, \tilde{X}_1(u)) = I_{\{Y_1=j\}}$, $h_{jk}(Y_i, Y_{i+1}, \tilde{X}_{i,i+1}(u)) = I_{\{Y_i=j, Y_{i+1}=k\}}$, let $med\ \tilde{X}_i = T_i$, then model (1) becomes

$$P(Y|X) = \frac{P(Y_1)}{Z_\theta(X)} \prod_{i=1}^{n} P(T_i|Y_i) \prod_{i=1}^{n-1} P(Y_{i+1}|Y_i), \quad (3)$$

$P(T_i|Y_i = j) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}d_i} exp\{-\frac{(T_i-b_j)^2}{2d_j^2}\}$, $Z_\theta(X) = \sum_Y P(Y_1) \prod_{i=1}^{n} P(T_i|Y_i) \prod_{i=1}^{n-1} P(Y_{i+1}|Y_i)$. Model (3) is equivalent to a HMM with normal emission distribution. In this regard, if model (1) is built base on median smoothed data $\{med\ \tilde{X}_i\}$, the model parameters and feature functions are selected as above, then the model (1) reduces to model (3). However, we notice that in our model (1), neither the initial function $l_j(Y_1, \tilde{X}_1(u))$ nor the transition function $h_{jk}(Y_i, Y_{i+1}, \tilde{X}_{i,i+1}(u))\}$ is a simple index function. They depend on the observation X. Moreover, the parameters θ of model (1) are with more freedom than that of model (3). These properties make our model (1) more promising.

References

1. Carter NP. Methods and strategies for analyzing copy number variation using dna microarrays. *Nat Genet* 2007; **39(7 Suppl)**: S16–21.
2. Chiang DY, Getz G, Jaffe DB, Kelly MJO, Zhao X, Carter SL, Russ C, Nusbaum C, Meyerson M, Lander ES. High-resolution mapping of copy-number alterations with massively parallel sequencing. *Nat Methods* 2009; **6(1)**: 99–103.
3. Cho EK, Tchinda J, Freeman JL, Chung YJ, Cai WW, Lee C. Array-based comparative genomic hybridization and copy number variation in cancer research. *Cytogenet Genome Res* 2006; **115(3-4)**: 262–272.
4. Colella S, Yau C, Taylor JM, Mirza G, Butler H, Clouston P, Bassett AS, Seller A, Holmes CC, Ragoussis J. Quantisnp: an objective bayes hidden-markov model to detect and accurately map copy number variation using snp genotyping data. *Nucleic Acids Res* 2007; **35(6)**: 2013–2025.
5. Freeman JL, Perry GH, Feuk L, Redon R, McCarroll SA, Altshuler DM, Aburatani H, Jones KW, Tyler-Smith C, Hurles ME, Carter NP, Scherer SW, Lee C. Copy number variation: new insights in genome diversity. *Genome Res* 2006; **16(8)**: 949–961.
6. Guha S, Li Y, Neuberg D. Bayesian hidden markov modeling of array CGH data. *J Amer Statist Assoc* 2008; **103(482)**: 485–497.
7. Hupe P, Stransky N, Thiery JP, Radvanyi F, Barillot E. Analysis of array cgh data: from signal ratio to gain and loss of dna regions. *Bioinformatics* 2004; **20(18)**: 3413–3422.
8. 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.
9. Hayes M and Li J. A linear-time algorithm for analyzing array CGH data using log ratio triangulation. *Lecture Notes in Bioinformatics* 2009; **5542** : 248-259.
10. 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.
11. Redon R, Ishikawa S, Fitch KR, Feuk L, Perry GH, Andrews TD, Fiegler H, Shapero MH, Carson AR, Chen W, Cho EK, Dallaire S, Freeman JL, Gonzalez JR, Gratacos M, Huang J, Kalaitzopoulos D, Komura D, MacDonald JR, Marshall CR, Mei R, Montgomery L, Nishimura K, Okamura K, Shen F, Somerville MJ, Tchinda J, Valsesia A, Woodwark C, Yang F, Zhang J, Zerjal T, Armengol L, Conrad DF, Estivill X, Tyler-Smith C, Carter NP, Aburatani H, Lee C, Jones KW, Scherer SW, Hurles ME. Global variation in copy number in the human genome. *Nature* 2006; **444(7118)**: 444–454.
12. 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)**: E431–439.
13. Shen F, Huang J, Fitch KR, Truong VB, Kirby A, Chen W, Zhang J, Liu G, McCarroll SA, Jones KW, Shapero MH. Improved detection of global copy number variation using high density, non-polymorphic oligonucleotide probes. *BMC Genet* 2008; **9:27** 1471–2156.
14. Snijders AM, Nowak N, Segraves R, Blackwood S,

Brown N, Conroy J, Hamilton G, Hindle AK, Huey B, Kimura K, Law S, Myambo K, Palmer J, Ylstra B, Yue JP, Gray JW, Jain AN, Pinkel D, Albertson DG. Assembly of microarrays for genome-wide measurement of dna copy number. *Nat Genet* 2001; **29(3)**: 263–264.

15. Snyman JA. Practical mathematical optimization : an introduction to basic optimization theory and classical and new gradient-based algorithms. Springer, New York, 2005.
16. Sutton C, McCallum A. An introduction to conditional random fields for relational learning. In: Getoor L, Taskar B (eds.), Introduction to Statistical Relational Learning. MIT Press, 2007.
17. Wang K, Li M, Hadley D, Liu R, Glessner J, Grant SF, Hakonarson H, Bucan M. Penncnv: an integrated hidden markov model designed for high-resolution copy number variation detection in whole-genome snp genotyping data. *Genome Res* 2007; **17(11)**: 1665–1674.
18. Willenbrock H, Fridlyand J. A comparison study: applying segmentation to array cgh data for downstream analyses. *Bioinformatics* 2005; **21(22)**: 4084–4091.