2012

Graphical model and algorithm for detecting DNA structural variation

Rong Shen
Iowa State University

Graphical model and algorithm for detecting DNA structural variation

by

Rong Shen

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Electrical Engineering

Program of Study Committee:
Zhengdao Wang, Major Professor
Sang W. Kim
Vivekananda Roy

Iowa State University

Ames, Iowa

2012

TABLE OF CONTENTS

LIST OF FIGURES iv

ABSTRACT v

CHAPTER 1. INTRODUCTION 1

1.1 Motivation 1

1.2 Contributions 2

CHAPTER 2. SYSTEM MODEL 4

2.1 Input Data Format 4

2.2 Single Base Pair Model 5

2.2.1 Proposed model 5

2.2.2 Likelihood Copy number 8

2.3 Hidden Markov model 9

2.3.1 Single-Step Dependency 9

2.3.2 Multi-Step Dependency 10

CHAPTER 3. ALGORITHM 12

3.1 Single-Symbol Processing 12

3.1.1 Copy number states 12

3.1.2 Computing Likelihood for $n = 1, 2, 3$ 13

3.1.3 Likelihood for large copy number 13

3.1.4 PAV detection 13

3.2 Factor graph and belief propagation algorithm 14

3.2.1 Factor graph 15

3.2.2 Sum-product algorithm 16
3.2.3 Sum-Product Algorithm for the CNV problem 17
CHAPTER 4. RESULTS, CONCLUSIONS, AND DISCUSSION 21
4.1 Results on maize data 21
4.1.1 Distribution of the reading depth 22
4.1.2 Processing per base pair 23
4.1.3 Message passing 23
4.1.4 Result filtering 24
4.1.5 Simulation verification 24
4.1.6 Processing measured genomic data 25
4.2 Conclusions, Discussion, and Future Work 27
4.2.1 Discussion 28
4.2.2 Future work 29
BIBLIOGRAPHY 30

LIST OF FIGURES

2.1	Single Base Pair Model	5
2.2	HMM Model	9
3.1	Model Factor Graph	15
3.2	Building Block of Sum-Product Algorithm	16
3.3	Sum-Product Algorithm for the CNV problem	17
3.4	Special Nodes	19
4.1	Reading depth distribution in both the reference and the sample	22
4.2	CNV simulation	24
4.3	PAV Simulation	25
4.4	PAV Segments	26
4.5	partial PAV Segments	26
4.6	CNV Segments 2	27

ABSTRACT

Next-generation sequencing (NGS) has revolutionized the detection of structural variation in genome. Among NGS strategies, reading depth is widely used and paramorphism information contained inside is generally ignored. We develop an algorithm that can fully exploit both reading depth and paramorphism information. We embed mutation procedure in our system model for estimating prior likelihood of single nucleotide base. Hidden Markov model is used to connect single base into segments and belief propagation algorithm is performed for the optimal solution of the HMM model. Simulations show promising results in detecting important types of structural variation. We have applied the algorithm on the maize B73 and MO17 genome data and compared the results with those obtained from arrayCGH method based micro-array data. Inconsistency between the two sets of data is discussed.

CHAPTER 1. INTRODUCTION

1.1 Motivation

In genome research, structural variation is defined as insertions, deletions and inversions in the sequence level or copy number variation (CNV) and presence/absence variation (PAV) in genomic level. Structural Variations (SV) are associated with the cause of disease as well as different traits between individuals [1–3]. Detecting structure variation in genomes has been under research quite long and great development has been achieved [4].

In the past, array-based method was used widely in SV detection and genotyping. Typical methods include array comparative genomic hybridization (arrayCGH) and SNP microarrays. Due to the resolution of probes, array-based methods have limitations in accuracy for longer CNV segments larger than 10 thousand base pairs (10 kbp) [5]. Another disadvantage is high-cost and intensive labor for re-sequencing arrays. Sequencing-based method, named next-generation sequencing (NGS), requires less labor and has less limitations in accuracy. Through *de novo* assembly, given long and accurate enough reading sequence, all kinds of SV could be reconstructed [4]. However, *de novo* assembly is still under development to reduce its complexity and improve algorithm speed as well as reduce cost for large genome datasets.

Tuzun et al. [6] proposed a way of detecting accurate small SV segments less than 1 kbp using paired-end reading (PEM) strategy. However, such method has little statistical power for SV larger than 1kbp due to PEM's limited resolution of detecting large segments. Small length SVs as well as their boundaries could be estimated precisely through exploiting paramorphism information. Zollner et al. [2] applied Bayesian computations and expectation-maximization (EM) algorithm to a known CNV location and achieved accurate estimation of CNV carrier status and its boundaries. However, known location is necessary in this algorithm. Compared

to them, reading depth provides a wide detection range and statistical accuracy. Event-Wise Testing achieved fast algorithm speed via processing intervals of reading depth and applied statistical significance test to the intervals with controlled significance level [7]. It achieved satisfactory results for CNV segments around 1000bp. A method called CNAseg applied Skellam distribution to reading depth and employed Hidden Markov Model (HMM) on combining segments of reading counts and found better estimation and precision besides lower significance level [3].

All the methods above have their own advantages and limitations. If we combine some of those methods, improved accuracy and detection range may be achieved. In this thesis, we are interested in combining both reading depth and paramorphism to achieve better CNV detection range and accuracy. Ideally, segments less or larger than 1 kbp can both be detected accurately.

1.2 Contributions

The main contributions of the thesis are the follows:

1. We propose a novel framework to model the process of nucleotide copies, paramorphism, and the randomness in the sequencing sampling process. Although the model is proposed for deriving the subsequent algorithm, it may also be useful for other investigations of related problems.

2. We evaluate likelihood of CNV considering mutation probability at each base pair location. The evaluation takes into account both reading depth and paramorphism information.

3. We proposes a simple hidden Markov model (HMM) relating the copy number variation and presence/absence variation of neighboring base pairs. We also derive a belief propagation algorithm for the specific problem to estimate the copy numbers and possible presence/absence variation.

The algorithm has been applied to lab measured data comprising a reference genome and a sample genome. The results are compared with those obtained with arrayCGH.

CHAPTER 2. SYSTEM MODEL

We describe in this chapter the system model for our proposed detection algorithm. The model consists two parts: 1) a single symbol model for a base pair that considers copy number variation, mutation, and the randomness in the sampling process; and 2) a Hidden Markov Model (HMM) that incorporates the dependence among neighboring base pairs. The algorithms for performing CNV detection using the proposed model and the measurement data will be described in the next chapter.

2.1 Input Data Format

Before we describe the model that we will build, we first describe the input data format for the problem. This information will help understand the nature of the problem, and the model that we will describe later.

The input is the assembled sequenced data stored in a PILEUP format [8]. An example of some lines of the file is shown below:

```
0 5896 C 4 .,,. IDII
0 5897 C 4 .,,. I.0I
0 5898 A 4 .,,. I%:I
0 5899 T 4 .,,. I&/I
0 5900 G 4 .,,. I2?I
0 5901 G 4 .,,. IIII
0 5902 G 4 .,,.  I.HI
0 5903 A 4 .,,. I+II
0 5904 G 4 .,,. II@I
0 5905 A 4 .g,. <&II
0 5906 A 4 .$,,. IIII
```

The first column specifies the chromosome number. The second column specifies the gene base pair index within the chromosome. The third column specifies the nucleotide on the ref-

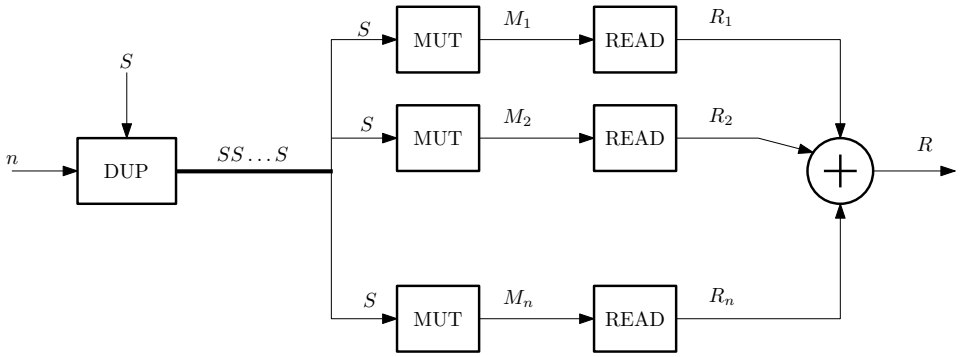

Figure 2.1 Single Base Pair Model

erence base. The fourth column specifies the reading depth of the sequencing output. The fifth column specifies the reading results, where matches are marked as dot and comma depending on whether the match with the reference base is exact match or reverse match. If not a match at a certain read, the result of read is explicitly marked. The last column specifies the mapping qualities encoded using ASCII characters.

Two such PILEUP files are available: one for the reference species and the other for the sample species. The problem is to process the sequencing data contained in the files and detect possible CNV and PAV in the sample species, as compared to the reference species.

2.2 Single Base Pair Model

We first propose a model that incorporates the possible copy number variation, mutation, and the randomness in the sampling process. In this model, we only consider the case where the ratio of the copy numbers n between the sample species and the reference species is larger than or equal to one. In other word, assuming the copy number of the reference is 1 at a certain base pair. The cases where the copy number of the sample is smaller than that of the reference will be considered separately later, in next chapter.

2.2.1 Proposed model

Our proposed model is depicted in Figure 2.1. The model describes what happens to a single base pair in the sample species. A symbol S goes through three steps to produce the observed

data: copying, mutation, and sampling (or reading). We are interested in using the model to estimate the likelihood of copy numbers from the known reading depth and the reading distribution, available in the PILEUP format data. We next describe the model in more details. There are 4 possibilities for S, namely $\{A, C, G, T\} := \mathcal{B}$.

1. *Copy.* For each symbol, assume its true copy number is n. The DUP block in the figure replicates the symbol S and produces n copies of S at its output.

2. *Mutation.* Each of the n copies can mutate to a different symbol with certain probability. This is represented as a MUT block in the figure. For simplicity, we assume the non-mutation probability for each copy of each symbol is the same as $(1-p)$. For example, if $S = A$, the probability for a symbol A to stay as A after the mutation block is $(1-p)$. With probability $p/3$, it can mutate to one of the other three possible nucleotides C, G, and T. Let $M_1, M_2, \ldots, M_n$ be the n symbols after mutation. We denote the type of the mutated symbols as $\mathbf{n} = (n_A, n_C, n_G, n_T)$, where n_i is the number of symbols i in the sequence $(M_1, M_2, \ldots, M_n)$, for $i \in \{A, C, G, T\}$.

   The mutation distribution $\mathbf{n}$ is a vector describing mutation distribution in the order of ATGC. For example, if $S = A$, and there is no mutation, then $\mathbf{n} = [n, 0, 0, 0]$. If A has one copy symbol mutated to G, then $\mathbf{n} = [(n-1), 0, 1, 0]$.

   Given the known symbol S, and the copy number n, the mutation distribution is multinomial:

$$\Pr(\mathbf{n}|n, S) = \frac{n!}{\prod_{i\in\mathcal{B}} n_i!} \prod_{i\in\mathcal{B}} P_i^{n_i}, \tag{2.1}$$

   where

$$P_i = \begin{cases} 1-p, & i = S, \\ p/3, & i \neq S. \end{cases} \tag{2.2}$$

3. *Reading.* Each mutated copy has a certain probability of being sampled by the sequencing procedure. The reading result has two parts of information: the reading depth, and the reading distribution.

Random distribution for the reading depth can be modeled by a Poisson distribution, a negative binomial distribution or a Skellam distribution. In this thesis, we use Poisson distribution to model the reading depth. Assuming the reading depth number is k, the probability of getting a known k reading depth given copy number n should be like this:

$$\Pr(k|n) = \frac{(n\lambda)^k e^{-n\lambda}}{k!} \tag{2.3}$$

where λ is the parameter of the Poisson distribution. Symbols in reference genome are assumed to have only one copy. If there is no copy procedure, sample symbol reading depths are expected to have the same mean as the reference symbols. So we will choose the reading depth in the reference genome times possible copy numbers as the parameter λ, the expected rate of increasing number for that symbol.

4. *Reading Error.* In the process of mapping symbols, there is a probability of error for each reading symbol. However, in this thesis, for simplicity, we consider the probability of error to be the same for all, represented by ϵ. There are two situations that both lead to detecting a copy for a typical symbol, for instances, A. If a symbol is indeed A, and it is read correctly, there would be one A in the final reading result. Another possibility is that the symbol was indeed one of C, G, or T and is mistakenly read into A. Suppose there are n_a counts of symbol A, n_G counts of G, n_C counts of C and n_T counts of T. For each symbol, the probability of detecting one A in the observed data should be:

$$q_A = (1-\epsilon)n_a/n + (\epsilon/3)(n_g + n_c + n_t)/n, \tag{2.4}$$

where $n = n_A + n_C + n_G + n_T$. In general, for any symbol i, the probability of reading an i would be $q_i = (1-\epsilon)n_i/n + (\epsilon/3)\sum_{j\neq i} n_j/n$. It could be written in matrix form:

$$\begin{bmatrix} q_A \\ q_C \\ q_G \\ q_T \end{bmatrix} = \begin{bmatrix} 1-\epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 1-\epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 1-\epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 1-\epsilon \end{bmatrix} \begin{bmatrix} n_A/n \\ n_C/n \\ n_G/n \\ n_T/n \end{bmatrix} \tag{2.5}$$

Knowing the distribution of the symbols in the final reading result, we deduce that the reading result distribution is multinomial. That is, given the mutation distribution $\mathbf{n}$,

the distribution of the reading results $\mathbf{k} := (k_A, k_C, k_G, k_T)$ should be as follows:

$$\Pr(\mathbf{k}|k, \mathbf{n}) = \frac{k!}{\prod_{i\in\mathcal{B}} k_i!} \prod_{i\in\mathcal{B}} {q_i}^{k_i} \tag{2.6}$$

where q_i is defined in (2.5), $\mathcal{B} = \{A, T, G, C\}$, and $k = \sum_{i\in\mathcal{B}} k_i$ is the reading depth.

2.2.2 Likelihood Copy number

Having derived the conditional probabilities in (2.1), (2.3), (2.5), we next derive the likelihood functions $\Pr(\mathbf{k}|n, S)$, which is the desired function for processing the input data for each base pair.

Using the chain rule of probability, we have

$$\Pr(\mathbf{k}, k, \mathbf{n}|n, S) = \Pr(\mathbf{n}|n, S) \cdot \Pr(k|n, S, \mathbf{n}) \cdot \Pr(\mathbf{k}|n, S, \mathbf{n}, k) \tag{2.7}$$

$$= \Pr(\mathbf{n}|n, S) \cdot \Pr(k|n) \cdot \Pr(\mathbf{k}|\mathbf{n}, k) \tag{2.8}$$

where in (2.8), we have used

$$\Pr(k|n, S, \mathbf{n}) = \Pr(k|n) \tag{2.9}$$

which states that the reading depth does not depend on what symbols are being read, and

$$\Pr(\mathbf{k}|n, S, \mathbf{n}, k) = \Pr(\mathbf{k}|\mathbf{n}, k) \tag{2.10}$$

which states that once the mutation distribution $\mathbf{n}$ is known, and for a given reading depth k, the reading distribution $\mathbf{k}$ does not depend on the copy number n and the source symbol S.

We can then marginalize (average out) the mutation $\mathbf{n}$ to obtain

$$\Pr(\mathbf{k}, k|n, S) = \sum_{\mathbf{n}} \Pr(\mathbf{k}, k, \mathbf{n}|n, S), \tag{2.11}$$

where the summation is over $\mathbf{n}$ such that $\sum_{i\in\mathcal{B}} n_i = n$. Further marginalizing the total reading depth k, which is trivial as $k = \sum_{i\in\mathcal{B}} k_i$ deterministically, we have

$$\Pr(\mathbf{k}|n, S) = \sum_{\mathbf{n}} \Pr(\mathbf{k}, k, \mathbf{n}|n, S) \tag{2.12}$$

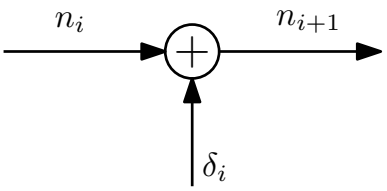

Figure 2.2 HMM Model

To summarize, the likelihood function can be obtained using (2.12), (2.8), (2.1), (2.3), (2.5), as follows:

$$\Pr(\mathbf{k}|n,S) = \sum_{\mathbf{n}} \left[\frac{n!}{\prod_{i\in\mathcal{B}} n_i!} \prod_{i\in\mathcal{B}} {P_i}^{n_i} \right] \cdot \left[\frac{(n\lambda)^k e^{-n\lambda}}{k!} \right] \cdot \left[\frac{k!}{\prod_{i\in\mathcal{B}} k_i!} \prod_{i\in\mathcal{B}} {q_i}^{k_i} \right] \tag{2.13}$$

where q_i's are given in (2.5)

With the known information, namely the reference symbol S, and the reading distribution $\mathbf{k}$, we can compute likelihood of its copy number. The largest likelihood would be the most possible copy number, based on the read information at a single base pair.

2.3 Hidden Markov model

In the previous section, we view each symbol independent from the other symbols in the genome. However, this is not true in real. Copy procedure usually involves a large segment, thus two adjacent symbols have a high probability of sharing the same copy number. We term the difference of location index between two nodes as *distance*. The longer distance between two symbols, the lower probability of their sharing the same copy number.

2.3.1 Single-Step Dependency

To model the dependency of the neighboring symbols' copy numbers, we assume a hidden Markov model(HMM) and use it for combining long sequence of copy number estimates. HMM has been previously used to model SNP detection [3, 9]. In this thesis , we propose a simple but novel HMM model is depicted in Figure 2.2.

In the figure, n_{i-1} and n_i represents the copy number at two adjacent location $i-1$ and i. The variable δ_i could be viewed as the hidden state at location i. The mapping relationship

between copy number and hidden variable is:

$$n_{i+1} = 1 + (n_i + \delta_i - 1) \mod m. \tag{2.14}$$

The m above stands for the number of states in the model. The reason that the model is hidden Markov is that the states δ_i are not directed observed.

The change δ_i has a probability distribution like follows:

$$\Pr(\delta = 0) = p, \quad \text{and} \quad \Pr(\delta \neq 0) = 1 - p, \tag{2.15}$$

where $\delta = 0$ represents no change in copy number, and $\delta \neq 0$ represents that that is a change. In the case where there is a change, we will assume that the probability $(1-p)$ is equally split among the cases where $n_{i+1} \neq n_i$. The probability p will be chosen according to desired gene CNV segment length. For example, for a length of 1000 base pairs, we may set $p = 0.999$.

The above Markov model corresponds to setting

$$\Pr(n_{i+1}|n_i, n_{i-1}, n_{i-2}, \ldots) = \Pr(n_{i+1}|n_i) \tag{2.16}$$

and

$$\Pr(n_{i+1} = n_i|n_i) = p, \quad \Pr(n_{i+1} \neq n_i|n_i) = 1 - p. \tag{2.17}$$

2.3.2 Multi-Step Dependency

It is also possible that the dependency between two symbols that are separated by a distance $d > 1$ needs to be calculated. We assume that a similar HMM holds where n_{i+1} would be replaced in general by n_{i+d}. The transition probability will be chosen according to the d-step transition probability of the single-step Markov chain described in Section 2.3.1. Specifically, let m denote the total number of states for each n_i. The transition probability of (2.17) can be described by the following matrix

$$T = \begin{bmatrix} p & \frac{1-p}{m-1} & \cdots & \frac{1-p}{m-1} \\ \frac{1-p}{m-1} & p & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{1-p}{m-1} \\ \frac{1-p}{m-1} & \cdots & \frac{1-p}{m-1} & p \end{bmatrix} \tag{2.18}$$

where we have assumed that the states are numbered from 1 to m, and the (i, j)-th entry of the matrix is set to $\Pr(n_{i+1} = i|n_i = j)$. The d-step transition matrix is simply T^d, for $d = 1, 2, 3, \ldots$. Since the matrix T is a circulant matrix, it is possible to obtain the closed form expression of T^d through its eigen-value decomposition, where the eigen-vectors are Fourier basis vectors. The m eigen-values are as follows:

$$1, \frac{mp-1}{m-1}, \ldots, \frac{mp-1}{m-1}. \tag{2.19}$$

As a result, it can be shown that

$$T^d = \begin{bmatrix} \alpha & \beta & \ldots & \beta \\ \beta & \alpha & \ddots & \vdots \\ \vdots & \ddots & \ddots & \beta \\ \beta & \ldots & \beta & \alpha \end{bmatrix} \tag{2.20}$$

where

$$\alpha = \frac{1}{m} + \frac{m-1}{m} \cdot \left(\frac{mp-1}{m-1}\right)^d, \tag{2.21}$$

$$\beta = \frac{1-\alpha}{m-1}. \tag{2.22}$$

That is, $\Pr(n_{i+d} = n_i|n_i) = \alpha$, and $\Pr(n_{i+d} \neq n_i|n_i) = 1 - \alpha$.

CHAPTER 3. ALGORITHM

In this chapter, we describe the details of our proposed algorithm. The algorithm consists two major steps. In the first step, each symbol is processed independently to obtain the likelihood for different copy numbers and presence/absence variations. In the second step, the single-symbol information is jointly processed using the hidden Markov model through a belief propagation algorithm.

3.1 Single-Symbol Processing

We first describe the processing details for a single base pair. The input is one input line in the PILEUP file data (the reading results of the sequencing output for one base pair). And the output is the likelihood of various copy number possibilities (states).

3.1.1 Copy number states

In many genomes such as the maize genome, large copy number is common. However, the CNV that we do have interest is relative small numbers, usually less than 3. For this reason, and to reduce the computation complexity, we consider the cases where the copy number n is larger than 3 jointly.

For each symbol of the sample, we compute the likelihood of the copy number n being in state $i \in \mathcal{A} := \{0, 1^-, 1, 2, 3, 3^+\}$, where

- $n = 0$ means deletion: the segment was present in the reference but not present in the sample.
- $n = 1^-$ means copy number reduction: the copy number in the reference is larger than 1, and the copy number in the sample is smaller than that in the reference.

- $n = 1, 2, 3$ means respectively that the copy number is 1, 2, and 3.
- $n = 3^+$ means that the copy number is larger than 3.

3.1.2 Computing Likelihood for $n = 1, 2, 3$

Given the observation data, which is in the form of the reading depth, reading distribution, we can compute the likelihood for each base pair according to the formula in (2.13). For small copy numbers n, enumerating all possibilities of $\mathbf{n} = (n_A, n_C, n_G, n_T)$ such that the sum of the entries is equal to n is feasible, and that is what we are going to use. However for large n, it becomes computationally unfeasible. As such we propose a simplified computation for large copy numbers, as detailed below.

3.1.3 Likelihood for large copy number

For simplicity and faster processing speed, we can ignore the mutation information when the reading depth is large enough. This is intuitively reasonable since when the reading depth is large, the information contained in the reading depth alone may be strong enough for estimating the copy number. Specifically, ignoring the reading distribution, and only using the reading depth, we have

$$\Pr(k|n \geq 4) = \frac{\sum_{i=4}^{\infty} \Pr(k|n = i) \Pr(n = i)}{\sum_{i=4}^{\infty} \Pr(n = i)}. \tag{3.1}$$

The above formula is the product of Poisson distribution and the prior distribution of copy number. It is clear that to compute (3.1), some prior distribution on the copy number n needs to be assumed. To obtain such prior distribution from the data, we can use the empirical distribution of the reading depth as an approximation of the prior distribution of the copy number.

3.1.4 PAV detection

Deletion in SV has two patterns. One is that the reference genome has a segment or a symbol while it disappears in the sample genome. If a segment is not observed in the sample, it could be due to a miss during the sampling process. However, if this is a long segment, this

probability would be low. The most possible situation is that this segment is totally deleted in the sample genome. In our method, we detect those deleted symbols by finding the region that exists only in reference genome and not in the sample genome. If we observe a missing base pair in the sample, then we will assign a high likelihood to the state $n = 0$, which corresponds to deletion, and a low likelihood to the other cases. If no such deletion detected, then we should set the likelihood of $n = 0$ to be really small number, accounting for outliers possibilities.

The second situation is that a symbol or a segment existed in both reference and sample. However, the reading depth in the reference for that region is much larger than that in the sample. It is also possible that this is due to a miss in the sampling process for the sample genome. The same as the situation above, if the region is long enough, there is a large probability that the number of this region is decreased. The likelihood assignment to the copy numbers $n = 1, 2, 3$ in this case should present a decreasing trend from $n = 1$ to $n = 3$. This is because, if we use $n = 1^-$ to represent this deletion case (reduction in copy number), the likelihood for $n = 1^-$ should be the largest. Thus, we use our computed likelihood according to (2.13) as well as the ratio between reference and sample as a judgment for this case in our method. If the likelihood is decreasing and the ratio is larger than a certain threshold (say 1.6), the likelihood for deletion would be set to a relatively large number say 0.1. If one of the two criteria is not satisfied, the likelihood should be set to a really small number to exclude outliers possibilities.

3.2 Factor graph and belief propagation algorithm

What we have obtained so far in Section 2.3.1 are $\Pr(\mathbf{k}_j|n_j, S_j)$ for each symbol j. However, nearby symbols are dependent on each other according to the HMM. Given HMM described in Section 2.3, our next step is to find the posterior probability of copy number given all the symbols' copy number likelihoods in the genome. Let N denotes the total length of a chromosome of interest, our observed data are $\Delta = (S_1, \ldots, S_N; \mathbf{k}_1 \ldots \mathbf{k}_N)$. Our goal is to estimate the copy numbers n_j for $j = 1, \ldots, N$. Specifically, we would like to compute the

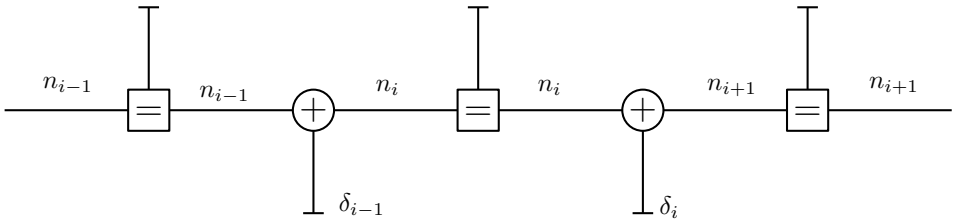

Figure 3.1 Model Factor Graph

following distributions:

$$P(n_j|S_1,\ldots,S_N;\mathbf{k}_1\ldots\mathbf{k}_N). \tag{3.2}$$

This problem in general has high complexity due to the need to marginalize all symbols but n_j in the joint posterior distribution of $(n_1,\ldots,n_N)$. For our problem, we have

$$P(n_1,\ldots,n_N|S_1,\ldots,S_N;\mathbf{k}_1,\ldots,\mathbf{k}_N) \tag{3.3}$$

$$\propto P(n_1,\ldots,n_N;\mathbf{k}_1,\ldots,\mathbf{k}_N|S_1,\ldots,S_N) \tag{3.4}$$

$$= P(n_1,\ldots,n_N|S_1,\ldots,S_N)\cdot P(\mathbf{k}_1,\ldots,\mathbf{k}_N|S_1,\ldots,S_N;n_1,\ldots,n_N) \tag{3.5}$$

$$= P(n_1,\ldots,n_N)\cdot P(\mathbf{k}_1,\ldots,\mathbf{k}_N|S_1,\ldots,S_N;n_1,\ldots,n_N) \tag{3.6}$$

$$= \prod_{i=1}^{N} P(n_i|n_{i-1}) \prod_{i=1}^{N} P(\mathbf{k}_i|S_i,n_i) \tag{3.7}$$

$$= \prod_{i=1}^{N} P(n_i|n_{i-1})P(\mathbf{k}_i|S_i,n_i) \tag{3.8}$$

where in (3.6) we have assumed that the copy numbers do not depend on the underlying symbols, and in (3.7) we have used the assumption that the copy numbers $(n_1,\ldots,n_N)$ form a Markov chain, and the fact that given the copy number n_i and the symbol S_i the reading result $\mathbf{k}_i$ at location i is independent of symbols, copy numbers, and reading results at other locations.

Due to the Markov structure, a low complexity algorithm for computing the posterior probabilities in (3.2) is possible through the belief propagation algorithm.

3.2.1 Factor graph

We can use factor graph [10] to represent the HMM constraints as a graphical model. Take any three adjacent symbols n_{i-1}, n_i, n_{i+1} for example. Their dependencies can be depicted as

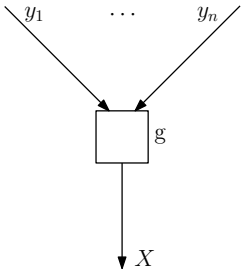

Figure 3.2 Building Block of Sum-Product Algorithm

in Figure 3.1. The factor graph describes the dependency of the variables. The half edges "⊣" describe input output variables. For more complete description of factor graph, see [10, 11].

3.2.2 Sum-product algorithm

In the tutorial paper [10], details about factors graph, sum-product algorithm and max-product algorithm are discussed. Here we stated briefly, in general, how the sum-product algorithm works. Given a factorization of a global function into a product of terms that are each a function of a subset of the variables, the sum-product algorithm efficiently computes the *summaries* of each variable, where a summary of variable x is a summation over all combinations of other variables but x of the global function. For example, if the global function is $f(x_1, x_2, x_3)$, then the summary of x_1 is

$$p(x_1) = \sum_{x_2, x_3} f(x_1, x_2, x_3). \tag{3.9}$$

The summary operation is the right operation needed to obtain a marginal distribution given the joint distribution of multiple random variables.

The sum-product algorithm simplifies the tasks of computing the summaries by using the fact that the global function can be factorized. The algorithm performs computations on the graphical model as follows. For each factor (square blocks), and for each edge of the factor, two messages are computed. One is leaving the block and one is entering the block. The message leaving the block is computed as a function of all the messages entering the block from all other edges connected to the block. Specifically, for a block g as shown in Figure 3.2, to obtain the outgoing message for x, we should multiply all messages coming on edges y_1 to y_n and sum

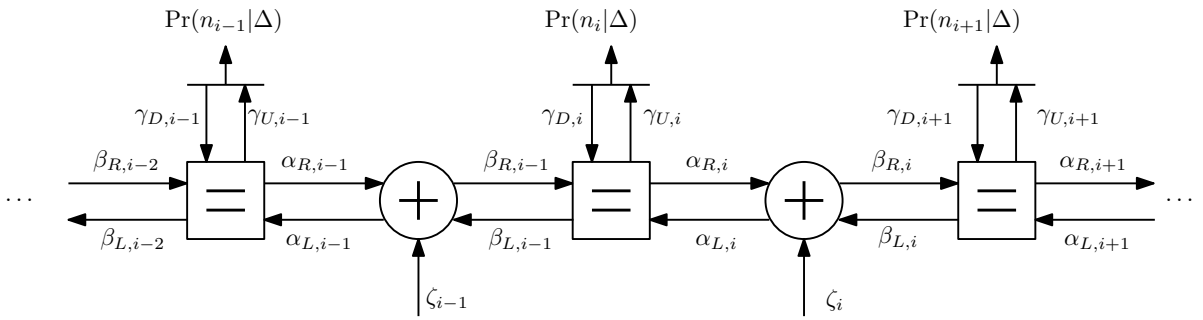

Figure 3.3 Sum-Product Algorithm for the CNV problem

them:

$$f_{g\to x}(x) = \sum_{y_1} \dots \sum_{y_n} g(x, y_1, \dots, y_n) \times f_{y_1\to g}(y_1) \times \dots \times f_{y_n\to g}(y_n). \tag{3.10}$$

This operation is performed on all blocks and all edges recursively. Due to the nature of the algorithm, it is often also referred to informally as *message-passing algorithm*.

3.2.3 Sum-Product Algorithm for the CNV problem

We get back to our problem. Due to the linear nature of our graphical model (the Markov chain), there should be bi-directional message flows. The HMM model for inverse direction is the same except for different transition probabilities. The algorithm diagram is shown in Figure 3.3, where it can be seen that for each edge of the HMM, there are two messages in opposite directions.

3.2.3.1 Initialization

We define the following notation for probabilities:

$$\alpha_{L,i}(n_i) \propto P(n_i|S_{i+1},\ldots,S_N;\mathbf{k}_{i+1},\ldots,\mathbf{k}_N) \tag{3.11}$$

$$\alpha_{R,i}(n_i) \propto P(n_i|S_1,\ldots,S_i;\mathbf{k}_1,\ldots,\mathbf{k}_i) \tag{3.12}$$

$$\beta_{L,i}(n_{i+1}) \propto P(n_{i+1}|S_{i+1},\ldots,S_N;\mathbf{k}_{i+1},\ldots,\mathbf{k}_N) \tag{3.13}$$

$$\beta_{R,i}(n_{i+1}) \propto P(n_{i+1}|S_1,\ldots,S_i;\mathbf{k}_1,\ldots,\mathbf{k}_i) \tag{3.14}$$

$$\zeta_i(n_i) = P(n_{i+1}|n_i) \tag{3.15}$$

$$\gamma_{D,i}(n) = P(\mathbf{k}_i|S_i,n_i) \tag{3.16}$$

$$\gamma_{U,i}(n) \propto P(n_i|S_1,\ldots,S_{i-1},S_{i+1},\ldots,S_N;\mathbf{k}_1,\ldots,\mathbf{k}_{i-1},\mathbf{k}_i,\ldots,\mathbf{k}_N) \tag{3.17}$$

The normalizing constants in the "$\propto$" parts are not important because the final result can always be normalized to 1 using the fact that a valid (conditional) probability mass function sums up to one. The algorithm will be initialized with

$$\beta_{R,0}(n) = \alpha_{L,N}(n) = p_i, \quad i \in \mathcal{A} \tag{3.18}$$

where p_i is the prior probability of a certain state i in the total state space $\mathcal{A}$ in the absence of any reading data.

3.2.3.2 Computation nodes

There are two types of nodes in Fig 3.3. One type that has the "=" sign represents constraints of equal value passing through the node. In mathematics expressions, it represents a factor in the global joint probability distribution of all variables in the form of

$$\delta(x-y)\delta(x-z).$$

Another node has + sign inside which represents constraints that one value equals to the summation of another two. It represents a factor of the form

$$\delta(z-x-y).$$

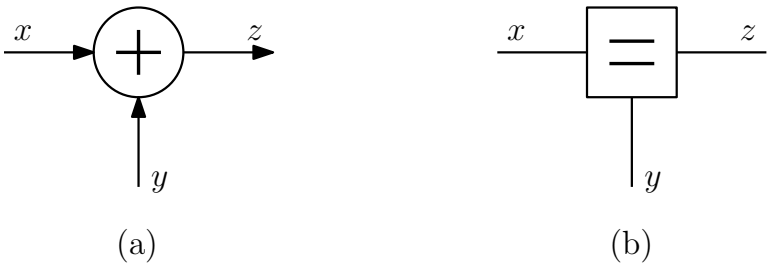

Figure 3.4 Special Nodes

Applying (3.10), we have the following computations to be performed in the message passing algorithm:

1. At the "Plus" nodes:

   (a) From left to right:

$$\beta_{R,i}(n) = \sum_{n_1,n_2} \delta(n - n_1 - n_2)\alpha_{R,i}(n_1)\zeta_i(n_2) = \sum_l \alpha_{R,i}(l)\zeta_i(n - l) \tag{3.19}$$

   (b) From right to left:

$$\alpha_{L,i}(n) = \sum_{n_1,n_2} \delta(n_1 - n - n_2)\beta_{L,i}(n_1)\zeta_i(n_2) = \sum_l \beta_{R,i}(l)\zeta_i(l - n) \tag{3.20}$$

2. At the "Equal" nodes:

   (a) From left to right:

$$\alpha_{R,i}(n) = \sum_{n_1,n_2} \delta(n - n_1)\delta(n - n_2)\beta_{R,i-1}(n_1)\gamma_{D,i}(n_2) = \beta_{R,i-1}(n)\gamma_{D,i}(n) \tag{3.21}$$

   (b) From right to left:

$$\beta_{L,i-1}(n) = \sum_{n_1,n_2} \delta(n - n_1)\delta(n - n_2)\alpha_{L,i}(n_1)\gamma_{D,i}(n_2) = \alpha_{L,i}(n)\gamma_{D,i}(n) \tag{3.22}$$

   (c) The message up:

$$\gamma_{U,i-1}(n) = \sum_{n_1,n_2} \delta(n - n_1)\delta(n - n_2)\alpha_{L,i}(n_1)\beta_{R,i-1}(n_2) = \alpha_{L,i}(n)\beta_{R,i-1}(n) \tag{3.23}$$

3.2.3.3 Output

The output of the algorithm is the posterior probabilities in (3.2). Specifically, we have

$$P(n_j|S_1,\ldots,S_N;\mathbf{k}_1\ldots\mathbf{k}_N)=\gamma_{D,i}(n)\gamma_{U,i}(n). \tag{3.24}$$

3.2.3.4 The whole algorithm

The algorithm will perform the computations in the following way:

1. The messages pointing to the right will be computed from left to right.
2. The messages pointing to the left will then be computed from right to left.
3. The messages pointing up will be computed in any order (say from left to right).
4. The output are computed by summing the $\gamma_{D,i}(n)$ and $\gamma_{R,i}(n)$ messages, for all i for all n.

CHAPTER 4. RESULTS, CONCLUSIONS, AND DISCUSSION

4.1 Results on maize data

We used maize genome data for analysis. Two sets of data for two species are available. The reference genome is named B73 and the sample one named MO17. All data is in pileup format.[1] Considering the large data, 5GB for both sample and reference genome, we can simplify the computation significantly with the following consideration. From our model in (2.13), it can be seen that in the case of no mutation, the prior likelihood of copy depends only on the sample reading depth. For the mutation case, this value depends on both reading depth and the distribution of the nucleotides in the reading results. Since the mutation probability is rather small, the computing the likelihood using depth-only information can significantly reduce the complexity.

We process the genome from chromosome to chromosome. For each chromosome, we read the index, symbol, reading depth and reading depth distribution from the sample genome and the index, and retrieve the reading depth from reference genome. We then transform the data from symbols to numbers. We use 1, 2, 3, and 4 in our algorithm to represent A T G C, respectively.

We find the nucleotide indices that exist in both sample and reference genome and assign the reference reading depth as the λ parameter for the corresponding symbol on the sample. The we take the union of the reference index and sample index. Thus if there is a total deletion, the corresponding value for λ is zero, which would be easy for the algorithm detection.

[1] Check http://samtools.sourceforge.net/pileup.shtml for details about pileup format.

4.1.1 Distribution of the reading depth

First, we obtained the distribution of the reading depth in the two genomes. This can give us some understanding of the distribution of the copy numbers.

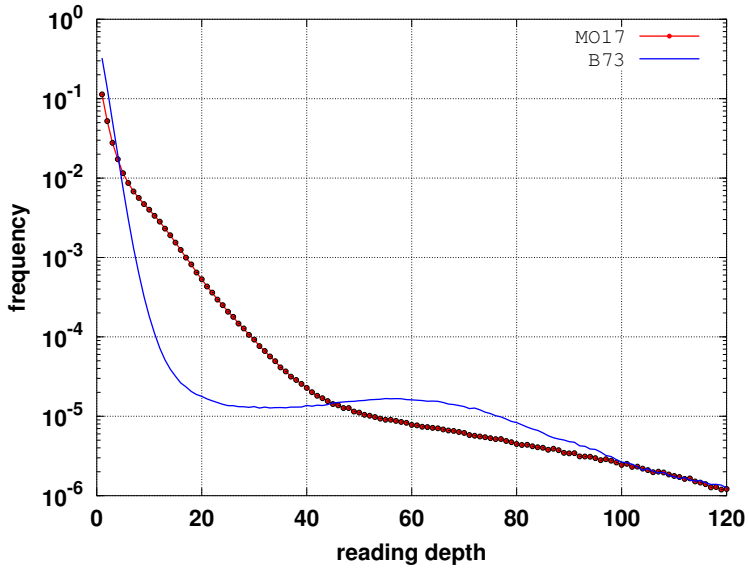

Figure 4.1 Reading depth distribution in both the reference and the sample

In Figure 4.1, we have shown the reading depth distribution in both the reference and the sample. For the sample (MO17) and for $n = 1, 2, 3$, the distribution can be approximated as $(0.41)^n$. For $n \geq 4$, we averaged the ratio between two adjacent numbers and found it can be approximated as $0.0637 \times (0.8544)^{(n-4)}$. Using this approximation, we can obtain

$$\begin{aligned} Pr(k|n \geq 4) &= \frac{\sum_{i=4}^{\infty} \frac{(i\lambda)^k e^{(-i\lambda)}}{k!} 0.0637(0.8544)^{(i-3)}}{\sum_{i=4}^{\infty} 0.0637(0.8544)^{(i-4)}} \\ &= \frac{\sum_{i=4}^{\infty} \frac{(i\lambda)^k e^{(-i\lambda)}}{k!} (0.8544)^i}{\sum_{i=4}^{\infty} (0.8544)^i} \\ &= 0.2732 \times \sum_{i=4}^{\infty} (i\lambda)^k e^{(-i(\lambda - \ln 0.8544))}/k! \\ &= 0.2732 \times \frac{\lambda^k (k+1)}{(\lambda - \ln 0.8544)^k} \sum_{i=4}^{\infty} (i\lambda - i \ln 0.8544)^k e^{(-i\lambda + i \ln 0.8544)}/(k+1)! \end{aligned} \quad (4.1)$$

The summation in (4.1) could be approximated as the cumulative distribution function(CDF) of gamma distribution. Thus the likelihood function finally turns out to be

$$L(n \geq 4) = 0.2732 \times \frac{\lambda^k (k+1)}{(\lambda - \ln 0.8544)^k} [1 - \gamma^{(inc)}(4\lambda, k)] \quad (4.2)$$

The $\gamma^{(inc)}(\cdot)$ represents the incomplete gamma integral from 1 to 4λ:

$$\gamma^{(inc)}(x,k) = \int_0^x t^{k-1}e^{-t}dt. \tag{4.3}$$

Use integral to replace the summation may lead to biased results. However, such bias would not affect the decision significantly.

4.1.2 Processing per base pair

At each base pair, we first determine whether this is a $n = 0$ case (no reading). If it is, we set the likelihood of deletion to a large number. If it is not and the sample reading depth, denoted as K, lies between $\frac{2}{3}\lambda$ to 10λ, we consider it as a calculable case. The next step is to judge if it is a mutation case or not. If there is no mutation, we obtain the likelihood values using depth-only information. This information can be stored in a precomputed table. If it is mutation case, we calculate the prior probability under the model described in the method part using (2.13). If the reading depth K is less than $\frac{2}{3}\lambda$, we set the likelihood value for the case of 1^- (copy number of reference larger than that of the sample) to a large value. For all the other case, set the likelihood has largest value in status 3^+ directly. Due to some error in the data, some extreme values in the likelihoods are removed.

4.1.3 Message passing

We then apply belief propagation to the prior probability from symbol to symbol, in the forward and backward directions. To reduce the amount that data that need to be stored in memory, we process the data in segments of length 100,000 symbols. After belief propagation on each block, we output the posterior probability results for all but the last 10,000 symbols. The next segment starts from the last 20,000 symbols of the previous segments and so on until the end of the data. This way, it requires less time for processing 10,000 symbols each time and the break points is also taken care of.

4.1.4 Result filtering

The copy number corresponding to the highest likelihood would be considered as the copy number for that point. There can be regions of copy numbers that are more than 1 but the length of the region is less than 1000 bp. We filter out the copy number variation segments that have lengths less than 1000 bp. The filtering procedure is important. Due to outliers in the data, small regions of CNV or PAV happen quite often, where the length could be as small as 20. Such regions are of little biological interest. Filtering out those small region would give us a cleaner result.

4.1.5 Simulation verification

Before applying the algorithm to the maize data, we performed an experiment by running the algorithm on simulated data first. The simulation data was generated based on the reference genome data. To test the detection capability of CNV from this algorithm, we doubled the reading depth for symbols whose indices are between 100,000 and 200,000, on chromosome number 0 in the reference genome. We took the first 400,000 nucleotides for simulation. The ideal result would be an interval of copy number equals to 2 from 100,000 to 200,000. The result from applying the algorithm is showed below: The CNV we detected starts on index

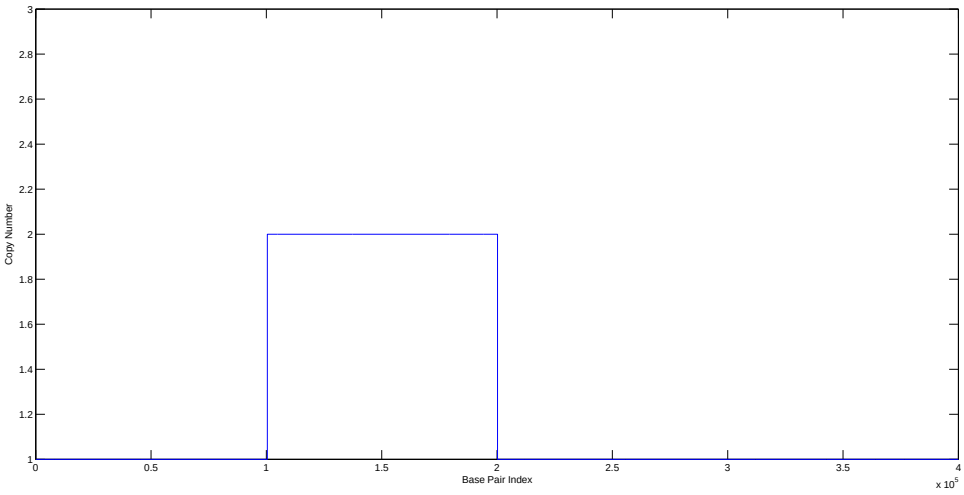

Figure 4.2 CNV simulation

100421 and ends on 199944 with values equal to 2, which almost fits our ideal figure. Before filtering, there are several small intervals equals to 3 in the figure. Those periods are caused by outlier values. We noticed that the outlier caused a small different region lasted about 10 symbols. After we applied a filter on it, those regions were smoothed.

For the deletion case, simulation data were generated in a similar way. We deleted data between 100,000 200,000 interval on chromosome number 0 in the B73 genome. Using the simulation data as sample genome and first 400,000 original data as reference genome, we would find the segments where deletion happened. The ideal figure should show an interval between 100,000 to 200,000 with copy number of 0. The results are showed below in Figure 4.3.

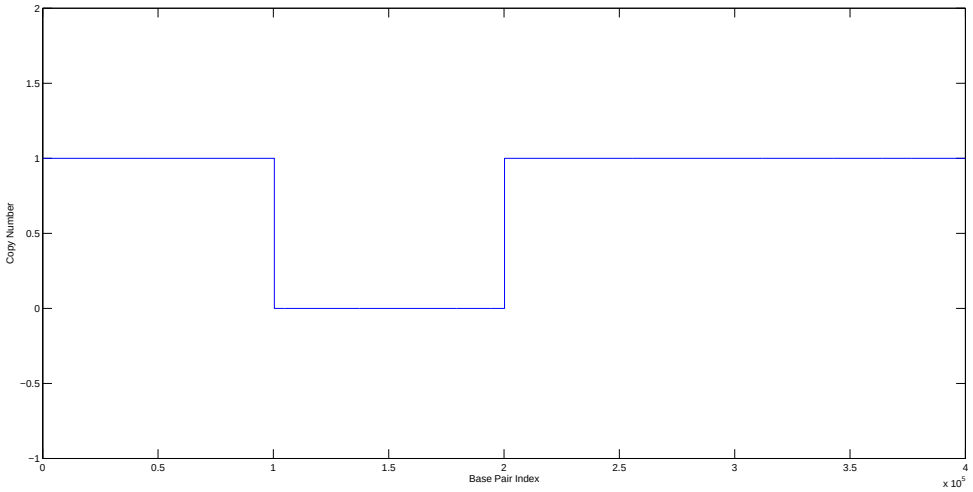

Figure 4.3 PAV Simulation

The PAV we detected starts on index 100421 and ends on 199944. Outliers are already smoothed in this case. The results almost fits the ideal figure.

Simulation test showed us this algorithm could detect both CNV and PAV correctly and their accurate position.

4.1.6 Processing measured genomic data

Now we move to the real situation, where there are more variable cases. We took chromosome number 6 in reference genome and its corresponding sample genome and ran our algorithm.

Since there are no ground truth data on the structure variation in the chromosome, we compared our result with that of micro-array-based method arrayCGH.

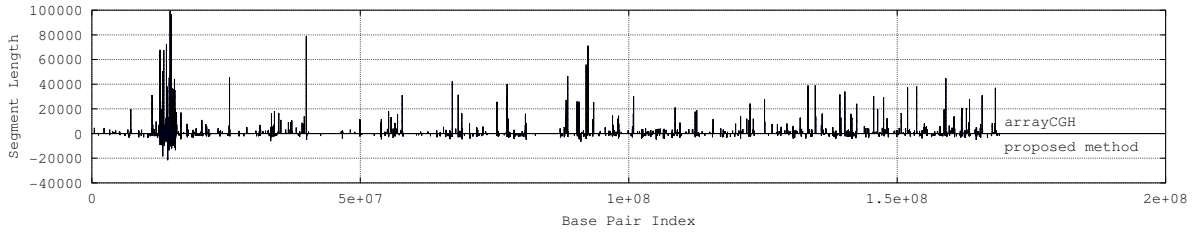

Figure 4.4 PAV Segments

We set the reading error probability to 10^{-3}. We only keep segments longer than 1kb in the output, by performing filtering on the message passing algorithm output.

In Figure 4.4 we plotted the detected PAV result, as compared with that from the arrayCGH method. The x axis denotes the starting point of a structure variation segment, and the length in the y direction describes the length of the segment. For the arrayCGH result, we plot them upward, and for our result, we plot them downward. It can be seen that most of the locations are overlapped in the figure. However, the lengths of the segments tend to be longer in the arrayCGH case.

There are a lot of small deletion segments in the data. Those small segments could be viewed as miss sampling. If we want to filter out a typical CNV segment, those small deletion segments would affect filter process in our algorithm. We fill those deletion segments as the CNV or PAV we want to as a way of smoothing data. By applying this method to detect status 6, partial deletion in sample, the following figure shows the result. The exact location

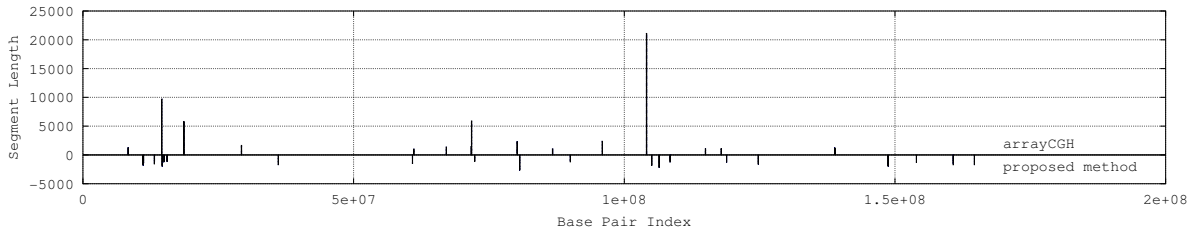

Figure 4.5 partial PAV Segments

is not fitted. However, segments all crowded around a similar range, which suggest this range

may suffer a partial deletion.

Except for the small segments of deletion, outliers would result in breakpoints. Fortunately, outliers usually affect a small area which lasts about 20 symbols. If we fill those regions as the CNV we want, it would be easy for the algorithm to detect CNV larger than 1kb. From the

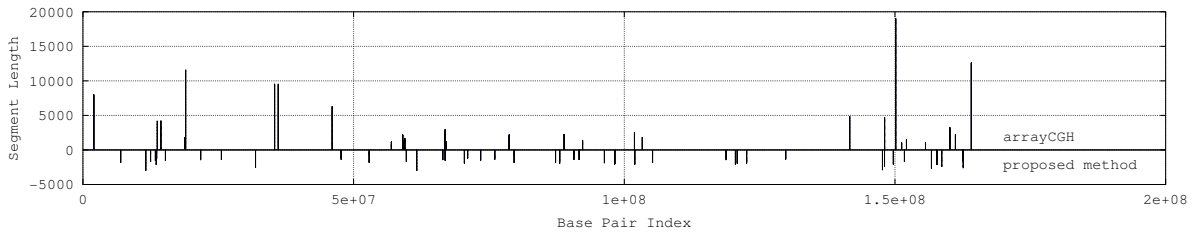

Figure 4.6 CNV Segments 2

figure, almost all the CNV indicated from arrayCGH is included in our result. Besides, our result provided more regions that CNV may happen.

4.2 Conclusions, Discussion, and Future Work

Using reading depth and paramorphism information, we developed a method for detecting copy number variation between two different genomes. We exploited the paramorphism information to strengthen reading depth power in detection of structural variation in genome. Also we applied belief propagation to solve the HMM and found the conditional single base copy number probability based on the prior information of other base pairs.

From the simulation test, we can see that outliers do have effect on the results. Outliers are common in the data. It is possible to find outliers before applying the message passing. However, this may increase program complexity. Considering single outlier could only result in small outstanding regions, it is possible to set a threshold length and smooth those regions. During the filter process, we set the threshold as 50b in size.

We applied Poisson distribution to describe sample reading depth distribution in the model. The constant λ in the Poisson distribution is taken directly from the reference reading depth, which assumed to be a constant in our model. However, reading depth from the reference genome is also sampled randomly, which resulted in a random distribution for λ and would

influence the prior likelihood in our model. Future work should focus on eliminating such fluctuation. The best way is to find the distribution of reading depth or find a way to minimize the variation of reference reading depth.

Our results offered similar regions and reliable likelihood of the PAV and CNV region compared with those detected from arrayCGH method. Besides, in the simulation, our method provides accurate start and end locations for simulated region.

4.2.1 Discussion

Paramorphism is important in this algorithm. It provides a larger probability for potential CNV region. Currently we do not have access to data with obvious paramorphism region. Assume a region of reading depth two (normalized by the Poisson parameter λ), and with mutations. Without mutation information, this region would be detected as CNV equal to 2, while in our method this region would be definitely larger than 2.

Results from simulation data suggest that this algorithm has the ability of correctly detecting CNV/PAV region. The start point and end point are close to the true points. Outliers would result in inaccurate detected copy number regions, which could be reasonably smoothed. The difference between the start/end point and true points could be due to the belief propagation algorithm.

Results from the maize data are hard to generate a conclusion. First of all, there is no true answer for the CNV/PAV region. The overlapped region could suggest that both our method and arrayCGH method detect this region as potential structural variation region. Two methods are consistent in those regions. The different regions may need further investigation if possible.

The bias in the start point and end point is caused by the sum-product algorithm. If there is a sudden change in the copy number for one base pair, it would last for some length until it stables. We can call it "cache region". Such phenomenon is unavoidable. From the simulation data, the average cache region is about 200b and the largest one is 421b, which are less than 1kb and would not affect the result too much.

For those overlapped regions, we can see the width from arrayCGH data is longer than the

width of our data. Our result just separated the long width into small segments. For example, the arrayCGH showed a width as long as 99315. In our results, this width has been separated into 11 smaller segments. Also some locations are not covered. This may be explained by the different resolution of two methods.

4.2.2 Future work

For the future work, several improvements could be done to the algorithm.

First, we chose the reference genome reading depth as the λ in the Poisson distribution. However, reference reading depth is also sampled randomly. The variation in the reference reading depth would influence the accuracy of our results. We may find a way to minimize the variance in the reading depth.

Besides, Poisson distribution does not work perfectly in the algorithm. The Poisson distribution requires the equality between its mean and variance. The mapping sequence cannot guarantee that this constraint is met. We may test some other distributions to find a better one.

When calculating the likelihood for larger copy number, we use reference reading depth to estimate the prior distribution of copy number. This also requires some adjustment.

Finally, the algorithm may need test on more data to validate its accuracy. It would be best if there more simulation data close to the real data but with known ground truth are available.

BIBLIOGRAPHY

[1] E. Gonzalez, H. Kulkarni, H. Bolivar, A. Mangano, R. Sanchez, G. Catano, R. J. Nibbs, B. I. Freedman, M. P. Quinones, M. J. Bamshad, K. K. Murthy, B. H. Rovin, W. Bradley, R. A. Clark, S. A. Anderson, R. J. O'Connell, B. K. Agan, S. S. Ahuja, R. Bologna, L. Sen, M. J. Dolan, and S. K. Ahuja, "The influence of CCL3L1 gene-containing segmental duplications on HIV-1/AIDS susceptibility," *Science*, vol. 307, no. 5714, pp. 1434–1440, Mar. 2005.

[2] S. Zöllner, G. Su, W. Stewart, Y. Chen, M. McInnis, and M. Burmeister, "Bayesian EM algorithm for scoring polymorphic deletions from SNP data and application to a common CNV on 8q24," *Genetic Epidemiology*, vol. 33, no. 4, pp. 357–368, 2009.

[3] S. Ivakhno, T. R. T, A. C. A, D. Evers, R. Cheetham, and S. Tavaré, "CNAseg — a novel framework for identification of copy number changes in cancer from second-generation sequecing data," *Bioinformatics*, vol. 26, no. 24, pp. 3051–3058, 2010.

[4] C. Alkan, B. Coe, and E. Eichler, "Genome structural variation discovery and genotyping," *Nature Reviews Genetics*, vol. 12, pp. 363–376, 2011.

[5] S. A. McCarroll, F. G. Kuruvilla, J. M. Korn, S. Cawley, J. Nemesh, A. Wysoker, M. H. Shapero, P. I. W. de Bakker, J. B. Maller, A. Kirby, A. L. Elliott, M. Parkin, E. Hubbell, T. Webster, R. Mei, J. Veitch, P. J. Collins, R. Handsaker, S. Lincoln, M. Nizzari, J. Blume, K. W. Jones, R. Rava, M. J. Daly, S. B. Gabriel, and D. Altshuler, "Integrated detection and population-genetic analysis of SNPs and copy number variation," *Nature Genetics*, vol. 40, no. 10, pp. 1166–1174, Oct. 2008. [Online]. Available: http://dx.doi.org/

[6] E. Tuzun, A. J. Sharp, J. A. Bailey, R. Kaul, V. A. Morrison, L. M. Pertz, E. Haugen, H. Hayden, D. Albertson, D. Pinkel, M. V. Olson, and E. E. Eichler, "Fine-scale structural variation of the human genome," *Nature Genetics*, vol. 37, no. 7, pp. 727–732, May 2005. [Online]. Available: http://dx.doi.org/10.1038/ng1562

[7] S. Yoon, Z. Xuan, V. Makarov, K. Ye, and J. Sebat, "Sensitive and accurate detection of copy number variants using read depth of coverage," *Genome Research*, vol. 19, no. 9, pp. 1586–1592, Sept. 2009.

[8] SAMtools: http://samtools.sourceforge.net/pileup.shtml.

[9] J. C. Marioni, N. P. Thorne, and S. Tavaré, "Biohmm: a heterogeneous hidden Markov model for segmenting array CGH data," *Bioinformatics*, vol. 22, no. 9, pp. 1144–1146, 2006.

[10] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 28–41, Jan. 2004. [Online]. Available: http://dx.doi.org/10.1109/MSP.2004.1267047

[11] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.