A HADOOP BASED GENOMIC STRUCTURAL VARIATION DETECTION PIPELINE

CHRIS WHELAN
CS 506 - PROBLEM SOLVING WITH LARGE CLUSTERS PROJECT REPORT

1. Background

Detection of structural variations involving large insertions, deletions, inversions, and duplications of genetic material is important in fields such as cancer research and evolutionary biology. It is possible to detect structural variation by generating paired short read sequences which are known to have come from regions of the sample DNA that are close to a given number of bases apart. The distance between the pairs in the sample is known as the *insert size*. When these read pairs are mapped to the reference genome, mapping locations that differ from the expected insert size potentially indicate the presence of a structural variation in the region of the reference spanned by the two alignment locations; see [1] for a review.

Most published approaches to finding structural variations from paired short read data use similar approaches: the pairs are mapped to the genome, and all pairs which mapped with a discordant insert size based on the expected distribution are then examined. These reads are then clustered to discover candidate breakpoints supported by a multiple discordant pairs. Sometimes an additional alignment step that examines all possible mappings for a pair is undertaken to identify pairs initially mapped as discordant that have a secondary concordant mapping.

I propose a new approach, in which all possible mappings for all read pairs are examined and allowed to contribute evidence for the presence or absence of structural variations in the region that they span. Using Hadoop/MapReduce, it is possible to handle the potentially large number of mapping locations for a pair, and to conduct location-by-location examination of the evidence for structural variations at each location in the genome.

2. Proposed Approach and Implementation Details

My pipeline consists of two consecutive MapReduce jobs and a post-processing step:

(1) MapReduce 1 - Alignments

- Map: Mappers examine each read from every pair individually, and determine all possible mapping locations for it using a mapper that can output multiple mapping locations and posterior probabilities for each location. In my case I am using the Novoalign aligner[6]. Each possible mapping is emitted under the identifying key for the read pair.
- Reduce: Reducers gather all mappings for each end of the read pair, and emit all possible paired mapping locations.

(2) MapReduce 2 - Structural Variation Scoring

- Map: Mappers examine individual pairs of alignment locations that were the output of MapReduce 1. For each read, they compute a structural variation score based on the insert size of the alignment and the probability of the mapping locations being correct. The genome is divided into 50bp windows, and for each window spanned by the alignment locations, the score is emitted with the index of the window as a key.
- Reduce: Reducers sum up all of the structural variation scores emitted for a given window location.

(3) Peak calling: The distribution of scores along the genome is examined and a peak-calling algorithm identifies regions with high scores relative to the background and calls them as likely to contain a structural variation.

Structural variation scores are computed as follows. For a given read pair P, which contains reads r_1 and r_2, Novoalign computes a set of alignment locations A_i. In addition, it computes a posterior probability for each alignment location, given statistics about the reference genome G and unmappable areas N as:

$$P(A|r,G) = \frac{P(r|A,G)}{P(r|N,G) + \sum_i P(r|A_i,G)}$$

Where the numerator represents the likelihood of the read given the alignment and the denominator contains terms for the likelihood that the read might map to an unmappable area of the genome as well as the sum of likelihoods of other possible alignment locations for the read. Given a pair of alignment locations $A = A_1, A_2$ for the two ends of a pair, I set the likelihood of the pair $P(A_1, A_2) = P(A_1|r_1, G) * P(A_2|r_2, G)$.

Based on the insert size s and the expected mean insert size μ and standard deviation σ, I assign a deletion score for the alignment pair based on the

likelihood of drawing an insert size from the distribution smaller than the observed insert size minus some factor α times σ:

$$P(deletion) = P(S \sim N(\mu, \sigma) < s - \alpha\sigma)$$

Based on initial hand-tuning I set a value of 1.5 for α. Each alignment pair then casts a vote v which is set to 1 if $P(deletion) > 0.5$ and -1 otherwise. Finally, the score emitted for the pair is given by:

$$S_A = v * P(A_1, A_2)$$

For each 50bp genomic window w, the reducers in MapReduce 2 then compute $\sum_{A_{spanning}} S_A$, where $A_{spanning}$ is the set of alignment pairs that span w.

Finally, for the peak calling portion of the algorithm, I adjust the scores by computing the average score for each location of the surrounding 50kb in the genome, and subtract that value from the score for the location. This adjusts for variations in the average scores in different genomic regions that are likely due to reference sequence features like the presence or absence of repetitive regions.

3. Preliminary Results

To evaluate my system, I created a simulated data set. Using simulated data allowed me to know with certainty the locations of the true structural variations in my sample. First, I added 75 structural variations to the human genome reference sequence for Chromosome 2, build hg19, using the SV_simulation tool provided in the PEMer[2] structural variation detection package. I chose chromosome 2 because it is the largest chromosome and the most representative of the entire human genome in terms of repetitiveness and other sequence features. The structural variations were inserted into the reference sequence at random locations, and were of the following types:

Type	Size	Number
Deletion	500	10
Deletion	1000	10
Deletion	5000	10
Deletion	10000	5
Insertion	500	10
Insertion	1000	10
Insertion	5000	10
Insertion	10000	5
Inversion	10000	5

I then generated 10,000,000 simulated read pairs from my modified reference sequence using SimSeq[3]. My read set attempted to simulate reads created with the Illumina Mate Pair protocol, with a target insert size of 3000bp and standard deviation in insert size of 300bp.

I compared the structural variations called by my system to the true structural variations generated by my simulation, as well as those called by the commonly used BreakDancer[4] and HYDRA[5] structural variation algorithms. I evaluated both programs based on an input of an alignment of the reads to the original hg19 reference sequence using Novoalign[6]. In the case of HYDRA, this was then followed by a secondary alignment using Novoalign with more sensitive parameter settings, producing multiple possible mappings per pair.

Leaving HYDRA and BreakDancer at their default settings, and setting a peak threshold of 325 for my system, the three algorithms were able to identify the following number of structural variations:

Algorithm	True Positives	False Positives
Hadoop SV Pipeline	24	122
HYDRA	28	52
BreakDancer	35	70

Based on the number of false positives, my algorithm is not quite competitive with the two published algorithms. On the positive side, however, my system did discover 7 structural variations that were not found by either HYDRA or BreakDancer, indicating that it may be able to detect certain types of structural variations that they cannot. In addition, I believe that with a better peak calling method, many more structural variations could be detected and false positive calls avoided, given a visual inspection of the scores in the true SV regions.

Computationally, the Hadoop/MapReduce framework makes this approach possible and convenient by allowing the distribution of the alignment task (the most computationally demanding part of the process) to multiple mappers. In addition, HDFS provides a good way to handle the large amounts of data produced by computing all possible alignment pairs for each read. This should allow this method to scale to larger, whole genome, data sets.

References

[1] Can Alkan, Bradley P Coe, and Evan E Eichler. Genome structural variation discovery and genotyping. *Nat Rev Genet*, Mar 2011.

[2] Jan O Korbel, Alexej Abyzov, Xinmeng Jasmine Mu, Nicholas Carriero, Philip Cayting, Zhengdong Zhang, Michael Snyder, and Mark B Gerstein. Pemer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. *Genome Biol*, 10(2):R23, Jan 2009.

[3] John St. John. Simseq. `https://github.com/jstjohn/SimSeq`.

[4] Ken Chen, John W Wallis, Michael D McLellan, David E Larson, Joelle M Kalicki, Craig S Pohl, Sean D McGrath, Michael C Wendl, Qunyuan Zhang, Devin P Locke, Xiaoqi Shi, Robert S Fulton, Timothy J Ley, Richard K Wilson, Li Ding, and Elaine R Mardis. BreakDancer: an algorithm for high-resolution mapping of genomic structural variation. *Nat Methods*, 6(9):677–81, Sep 2009.

[5] Aaron R Quinlan, Royden A Clark, Svetlana Sokolova, Mitchell L Leibowitz, Yujun Zhang, Matthew E Hurles, Joshua C Mell, and Ira M Hall. Genome-wide mapping and assembly of structural variant breakpoints in the mouse genome. *Genome Res*, 20(5):623–35, May 2010.

[6] Novocraft Inc. Novoalign. `http://www.novocraft.com`.