1B-6

Improvement of the Accuracy of Mapping by Composing Alleles

Kayo Okuda[1], Yoichi Takenaka[2], Tomoshige Ohno[2], Shigeto Seno[2], and Hideo Matsuda[2]
[1] Department of Information and Computer Science, School of Engineering Science, Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka, JAPAN
[2] Department of Bioinformatic Engineering, Graduate School of Information Science and Technology, Osaka University, 1-5 Yamada-Oka, Suita, Osaka, JAPAN

Background

Genome analyses using short reads which are generated by the high-throughput sequencer begin by mapping reads to the target genome. Therefore, they are sensitive to mapping results and high accuracy mapping is crucial. Ideal mapping results correctly identify the original location of each read. However, conventional mapping tools cause multireads and unmapped reads, that is, the mapping accuracy is insufficient. One reason for causing those reads is that only one genome sequence is used as the reference, even if the target organisms are polyploidy. That means both alleles are not treated equally and reads originating from a non-reference allele are more susceptible to mismapping when aligned to the reference allele, which contains at least one mismatch (in the case of SNPs) or gap (in the case of indels) [1].

Methods

We propose a novel mapping method that takes alleles into account. The inputs are a sequence data set *Read* of *l*-mer short reads and the reference genome *Ref*. Using these inputs, allele sequences are constructed and *Read* is remapped to the allele sequences. The output is the remapping result *Remap*. This concept is presented in Figure 1. In order to explain our method, we use the following functions:

- *classify(m)*: classify the mapping result into three classes: *uniq*,*multi* and *unmap*, which is a set of unique, multi and unmapped reads.
- *map(r, Ref)*: map read *r* to a reference sequence *Ref*.
- *detect(p, Rs, Ref)*: query whether a position *p* is a SNP using read set *Rs* and reference sequence *Ref*. If *p* is a SNP, this function returns *snp*. If *p* is not a SNP, this function returns *notsnp*.
- $base(m, p)$: query which types of bases the read of mapping result *m* has at position *p*. If $base(m, p)$ is A, T, G, or C, then the returned value is *a*, *t*, *g*, or *c*, respectively. If *m* does not include *p*, then *NULL* is returned.

Step 1: Map reads set to the reference genome

Read is mapped to the reference genome *Ref*.

$$Firstmap = map(Read, Ref)$$

The mapping result is classified into three subsets:

$$Fm_{uniq} = \{\, m \mid m \in Firstmap \,\&\, classify(m) = uniq \,\}$$
$$Fm_{multi} = \{\, m \mid m \in Firstmap \,\&\, classify(m) = multi \,\}$$
$$Fm_{unmap} = \{\, m \mid m \in Firstmap \,\&\, classify(m) = unmap \,\}$$

Step 2: Detect SNP positions

We use an SNP detection tool given two input data sets, Fm_{uniq} and *Ref*.

$$S_{pos} = \{\, s \mid s \in all\ positions \,\&\, detect(s, Fm_{uniq}, Ref) = snp \,\}$$

The set S_{pos} of detected SNP positions satisfies the following condition:

$$s_i < s_j \text{ for all } s_i, s_j \in S_{pos}\ (0 \le i < j < |S_{pos}|)$$

Step 3: Construct allele sequences

This step is composed of the following four sub-steps.

1) Classify the SNP positions

According to read length *l*, S_{pos} is classified into S_i which is SNP position subsets. Because, If adjacent s_i and s_{i+1} are separated by more than *l*, the base pairs of (s_i, s_{i+1}) cannot be determined. A set of S_i is S_{set}.

$$S_i = \{\, s_j, s_{j+1} \mid s_j, s_{j+1} \in S_{pos} \,\&\, (s_{j+1} - s_j) \le l \,\}$$
$$S_{set} = \{\, S_0, S_1, S_2, \ldots \mid S_i \cap S_j = \emptyset \}$$

2) Generate a graph for each subset of SNP positions

A weighted directed graph $G_i = (V_i, E_i, c_v, c_e)$ is generated for a subset $S_i = \{\, s_j, s_{j+1}, \ldots, s_{|s_i|-1} \,\}$ using Fm_{uniq}. An overview of this step is shown in Figure 2.

$$V_i = \{\, v_{j(b)} \mid s_j \in S_i \ \&\ base(m, s_j) = b \,\}$$
$$E_i = \{\, (v_{j(b_1)}, v_{j+1(b_2)}) \mid v_{j(b_1)}, v_{j+1(b_2)} \in V_i \,\&\, c_e((v_{j(b_1)}, v_{j+1(b_2)})) > 0 \,\}$$
$$c_v(v_{j(b)}) = \frac{|\{\, m \mid m \in Fm_{uniq} \,\&\, base(m, s_j) = b \,\}|}{|\{m \mid m \in Fm_{uniq} \,\&\, base(m, s_j) \neq NULL\}|}$$
$$c_e(e) = \frac{|\{m \mid m \in Fm_{uniq} \,\&\, base(m, s_j) = b_1 \,\&\, base(m, s_{j+1}) = b_2\}|}{|\{\, m \mid m \in Fm_{uniq} \,\&\, base(m, s_j), base(m, s_{j+1}) \neq NULL \,\}|}$$

If there is a mapping result *m* with base *b* at position s_j, then a node $v_{j(b)}$ is generated. The weight of the node is the ratio of *m* with base *b* at position s_j to *m* covering position s_j. If there is a mapping result *m* with both b_1 at s_j and b_2 at s_{j+1}, then a left–right direction edge $(v_{j(b_1)}, v_{j+1(b_2)})$ is generated. The

weight of this edge is the ratio of m with both b_1 at s_j and b_2 at s_{j+1} to m covering both s_j and s_{j+1}.

3) Search two adopted paths for each graph
The permutation of nodes on a path corresponds to the permutation of SNP position bases of one allele sequence. Then two adopted paths are searched in each graph. The searching method is to assign a score to each path and to select the 1st and 2nd largest score paths. The score of a path

$$p = (v_{i(b_i)}, v_{i+1(b_{i+1})}, \dots, v_{j-1(b_{j-1})}, v_{j(b_j)})$$

is defined as follows:

$$score(p) = \prod_{k=i}^{j} w(v_{k(b_k)}) \prod_{k=i}^{j-1} w\big((v_{k(b_k)}, v_{k+1(b_{k+1})})\big)$$

4) Construct allele sequences
The 1st and 2nd largest score paths are

$$p_1 = \left(v_{i_1(b_{i_1})}, v_{i_1+1(b_{i_1+1})}, \dots, v_{j_1-1(b_{j_1-1})}, v_{j_1(b_{j_1})}\right)$$
$$p_2 = \left(v_{i_2(b_{i_2})}, v_{i_2+1(b_{i_2+1})}, \dots, v_{j_2-1(b_{j_2-1})}, v_{j_2(b_{j_2})}\right)$$

then, the two permutations of SNP position bases are defined as follows:

$$ale_{i_1} = \left(b_{i_1}, b_{i_1+1}, \dots, b_{j_1-1}, b_{j_1}\right)$$
$$ale_{i_2} = \left(b_{i_2}, b_{i_2+1}, \dots, b_{j_2-1}, b_{j_2}\right)$$

After determining all base patterns of $S_{set} = \{S_0, S_1, S_2, \dots\}$, using the reference genome *REF*, two allele sequences are constructed. The SNP position bases of *REF* are replaced to $ale_{i_1}(0 \le i < |S_{set}|)$, and the corresponding sequence is labeled REF_1. In the same manner, REF_2 is constructed using $ale_{i_2}(0 \le i < |S_{set}|)$.

Step 4: Remap a sequence data set to allele sequences

Read are classified into three categories using the first mapping result, Fm_{uniq}, Fm_{multi}, and Fm_{unmap}.

$$R_{uniq} = \{\, r \mid m \in Fm_{uniq} \,\&\, m = map(r, Ref) \,\}$$
$$R_{multi} = \{\, r \mid m \in Fm_{multi} \,\&\, m = map(r, Ref) \,\}$$
$$R_{unmap} = \{\, r \mid m \in Fm_{unmap} \,\&\, m = map(r, Ref) \,\}$$

Then, R_{multi} and R_{unmap} are remapped to two allele sequences, REF_1 and REF_2.

$$Remap = map\left(R_{multi} \cup R_{unmap},\ REF_1 \cup REF_2\right)$$

Results

We performed two experiments in order to verify the effectiveness of our method. RNA-seq of human female (ERS025099) and a human genome (hg19) are used as the input. We used *Bowtie* mapping tool [2] and the *SAMtools* SNP detection tool [3]. Bowtie ran with two parameter '-k 2' and '-sam'. SAMtools ran under the default condition.

Experiment 1: Count the number of SNPs and graphs
This experiment aimed to verify the constructed allele sequences. First, we mapped RNA-seq to the human reference genome using *Bowtie* and classified the mapping results. The number of unique reads, multireads and unmapped reads were 48,809,928, 13,266,636, 15,118,696 respectively. Second, we detected the SNP positions using *SAMtools*. The number of detected SNP positions was 9,004 (coverage $\geq$ 30). Third, we generated graphs that correspond to the detected SNP positions. As a result, 8,151 graphs were created. This result indicates that proposed method is able to construct allele sequences because it detected 9,004 SNP positions and 710 graphs had two or more SNP positions were generated.

Experiment 2: Remap reads to two allele sequences
This experiment aimed to demonstrate that our method is more accurate than conventional methods. Using 8,151 graphs in the Experiment 1, we constructed two allele sequences and remapped multireads and unmapped reads to two allele sequences. The result shows 219,453 (0.28%) reads were additionally uniquely mapped. The result indicates that our method can reduce non-uniquely mapped reads, and consequently it contributes to more accurate mapping.

Conclusions

We proposed a method that takes alleles into account. In the experiments our method constructed allele sequences and remapped reads to those sequences. As a result, multireads and unmapped reads were reduced, that is, mapping accuracy could be improved.

References

[1] Rozowsky J, Abyzov A, Wang J, Alves P, Raha D, Harmanci A, …, Gerstein M: **AlleleSeq: analysis of allele-specific expression and binding in a network framework.** *Molecular System Biology* 20117:522

[2] Langmead B, Trapnell C, Pop M, Salzberg SL: **Ultrafast and memory-efficient alignment of short DNA sequences to the human genome.** *Genome Biology* 2009 **10**(3):R25

[3] Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, …, Durbin R; **The Sequence Alignment/Map format and SAMtools.** *Bioinformatics* 2009 **25**(16):2078-9

Figure 1 Overview of the proposed method