Supplementary Materials
Next Generation VariationHunter:
Combinatorial Algorithms for Transposon Insertion Discovery

Fereydoun Hormozdiari [1], Iman Hajirasouliha[1], Phuong Dao[1], Faraz Hach[1], Deniz Yorukoglu[1],Can Alkan[2], Evan E. Eichler [2], S.Cenk Sahinalp [1]

[1] School of Computing Science, Simon Fraser University, Burnaby, BC, Canada

[2] Department of Genome Sciences, University of Washington, and

April 29, 2010

1 Conflict Rules in Haploid Genome

The current state of art in identification of structural variations in sequenced genome using short read technologies reveals that identifying the SVs which share break-points or happen within a small distance from one another (e.g. twice as the size of paired-end insert-size) in the reference genome is a very challenging task. Throughout our methods, for each SV cluster, we define a segment in the reference genome as the *conflict zone* of the SV cluster and assume that two different SV clusters do not overlap in their conflict zones in a *haploid* genome sequence.

In order to define the conflict segment of an SV cluster, we will first give some definitions. For each SV cluster $VClu$, we define $minL(VClu)$, $maxL(VClu)$, $minR(VClu)$, $maxR(VClu)$ as following:

$$\begin{array}{rcl} minL(VClu) & = & min\{L_l(a_jpe_i)|a_jpe_i \in VClu\} \\ maxL(VClu) & = & max\{L_r(a_jpe_i)|a_jpe_i \in VClu\} \\ minR(VClu) & = & min\{R_l(a_jpe_i)|a_jpe_i \in VClu\} \\ maxR(VClu) & = & max\{R_r(a_jpe_i)|a_jpe_i \in VClu\} \end{array}$$

Next, for each valid cluster $VClu$ supporting different types of variations, we define the conflict zone $CZ(VClu)$ as following:

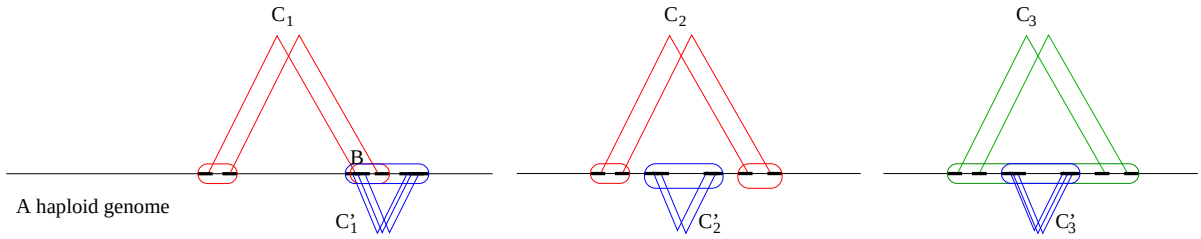
$$\text{CZ(Vclu)} = \begin{cases} [minL(VClu), maxR(VClu)] & \text{Del.} \\ [minL(VClu), maxR(VClu)] & \text{Ins.} \\ [minL(VClu), maxL(VClu)] & \text{Inv.} \\ \cup\, [minR(VClu), maxR(VClu)] & \\ [minR(VClu), maxR(VClu)] & \text{Copy from Left} \\ [minL(VClu), maxL(VClu)] & \text{Copy from Right} \end{cases}$$

Note that the conflict zone of an SV cluster which supports an inversion is split to two non-overlapping conflicting zones.

In Figure 1(a), two different SV clusters [1], C_1 and C_1' are seen where C_1 represents an *inversion* occurring in a block of a haploid genome sequence and C_1' a *insertion*. Let B (see Figure 1(a)) be a block where the *ends* of the mate-paired reads suggesting the inversion event location. As it can be seen, in Figure 1(a), C_1 and C_1' are conflicting SV clusters since both of them cannot occur at the same time. Figure 1(b) presents two SV clusters, C_2 and C_2', one representing an insertion and the other one representing an inversion. In this case, C_2 and C_2' can indeed be valid SV

[1] We remind the reader that each SV cluster represents a set of discordant mate-paired reads supporting exactly one particular SV.

clusters at the same time and are not conflicting with each other. Another example is shown in Figure 1(c) where SV clusters C_3 and C'_3 represent an insertion and a deletion event and conflict with each other.



(a) Conflicting: C_1 is an inversion cluster, shown in red together with its conflict zone, and C'_1 is an insertion cluster, shown in blue. (b) Not Conflicting: C_2 is an inversion cluster and C'_2 is an insertion cluster. (c) Conflicting: C_3 is a deletion cluster and C'_3 is an insertion cluster.

Figure 1: (a) shows two SV clusters together with their conflict zones. The conflict zones share a common block B in a haploid genome sequence; thus we consider them as conflicting SV clusters. (b) C_2 and C'_2 support an inversion event and an insertion event. Their conflict zones are not intersecting; thus C_2 and C'_2 are not considered conflicting. (c) shows two SV clusters that are conflicting in a haploid genome sequence, since $CZ(C_3) \subset CZ(C'_3)$.

2 Computational complexity of MPSV-CR

We will prove that MPSV-CR is NP-hard even if we have any positive weight on the cardinality of SC' and any positive penalty for unmapped reads (i.e. minimizing the function $g(SC') = k|SC'| + l\sum_{pe\in R}\delta(SC', pe)$ for some $k > 0$ and $l > 0$ is NP-hard.). When $l \geq k > 0$ (we denote this Case 1), minimizing $g(SC') = k|SC'|+l\sum_{pe\in R}\delta(SC', pe)$ is the same as minimizing $g(SC') = |SC'| + l'\sum_{pe\in R}\delta(SC', pe)$ where $l' = l/k$. When $k > l > 0$ (we denote this Case 2), minimizing $g(SC') = k|SC'| + l\sum_{pe\in R}\delta(SC', pe)$ is the same as minimizing $g(SC') = k'|SC'| + \sum_{pe\in R}\delta(SC', pe)$ where $k' = k/l$.

We prove that the MPSV-CR problem is NP-hard by using a reduction from the minimum set cover problem. Given C, a collection of subsets of a finite set S ($|S| = n$), we would like to find a subset $C' \subseteq C$ with the minimum cardinality such that every element in S belongs to at least one member of C'. Without loss of generality, we can assume Given an instance of the set cover problem, we build an instance of MPSV-CR as follows:

Case 1 ($k = 1$ and $k \leq \ell$): For each element $S_i \in S$, there is a discordant paired-end read pe_i, and corresponding to each set $C_j \in C$ we have a cluster $VClu_j = C_j$. We define $R = S$, $V(CG) = V(G)$, and $E(CG) = \emptyset$. It is easy to see that if we have a set cover C' of size $\leq t$, we can select a satisfiable set of clusters SC such that $g(SC) \leq t$. On the other hand, if we can select a satisfiable set of clusters SC such that $g(SC) \leq t$ which includes x clusters and y *uncovered* discordant paired-end reads, we can have a corresponding solution $C'' \subseteq C$ for the set cover instance with $|C''| \leq x + y \leq t$ by choosing at most y more sets to cover y uncovered elements.

Case 2 ($\ell = 1$ and $\ell < k$): We denote $p = \lceil k \rceil$. For each element $S_i \in S$, there are p discordant paired-end reads $pe_i, pe_{i+n}, \ldots, pe_{i+n(p-1)}$ and corresponding to each set $C_j \in C$ there is a cluster $VClu_j = \{pe_k | S_{k \bmod n} \in C_j\}$. We define $R = S$, $V(CG) = C$, and $E(CG) = \emptyset$ like in Case 1. If we have a set cover C' of size $\leq t$, we can select a set of clusters SC such that $g(SC) = kt$. When we can select a satisfiable set of clusters SC such that $g(SC) \leq kt$ which includes x clusters with y uncovered discordant paired-end reads. By the construction, we have $g(SC) = kx + y$ and y uncovered discordant reads correspond to $y' = y/p$ elements in S. And since $k(x + y') \leq kx + py' \leq kx + y \leq kt$, we have $x + y' \leq t$. Thus, it is similar to Case 1 the collection C'' of x sets and at most additional y' sets to cover y' uncovered elements is a solution to the set cover instance with cardinality less than or equal to t.

In the following, we show an inapproximability result even when we deal with a haploid genome.

Theorem 1. *There is no constant $\epsilon > 0$ for which MCSV-CR on haploid genome can be approximated within a factor of $n^{1-\epsilon}$ in polynomial time, unless $P = NP$.*

Proof. We use an approximation preserving reduction [1] from the Minimum Independent Dominating Set (MIDS) problem. Given a graph $G = (V, E)$ where $|V| = n$, the MIDS problem asks for a set $S \subset V$ with the minimum cardinality such that S is not only a dominating set but also an independent set of G. S is a dominating set of G if for each $v \in V$, either $v \in S$ or v is adjacent to some $v' \in S$. S is an independent set of the graph G if $\forall e \in E$: $e \nsubseteq S$.

Given an instance of the MIDS problem, we build an instance of MPSV-CR as following: Corresponding to each vertex v_i, we have a cluster $VClu_i$. We also set $MC = R = V(G)$, $V(CG) = V(G)$, and $E(CG) = E(G)$. Now for each $i \leq n$, we define the SV cluster $VClu_i = \{v | \exists e \in E : e = v_i v\}$ i.e. $VClu_i$ consists of vertices that are adjacent to v_i and includes v_i.

It is easy to see that the size of minimum independent dominating set of G is the same as of $f(SC_{OPT})$ where SC_{OPT} is the optimal satisfiable set of clusters. In general, given a satisfiable set of clusters SC, we can obtain the corresponding indepedent dominating set in G with the size less than or equal to $f(SC)$. This can be easily done by obtaining another satisfiable set of clusters SC' where all the reads are mapped such that $f(SC') \leq f(SC)$. Then we obtain the corresponding independent dominating set in G with the size equal to $f(SC')$. Hence, if MPSV-CR has an ϵ-approximation algorithm ($\epsilon > 1$) with polynomial running time, the MIDS problem also has a polynomial time approximation algorithm within the same factor.

However the MIDS problem does not have any polynomial approximation algorithm within a factor of $n^{1-\epsilon}$ for any $\epsilon > 0$ unless $P = NP$ [2]. Thus, the MPSV-CR problem is not likely to have a polynomial approximate algorithm within the same factor.

□

3 Simple Simulation

We pick known large deletions (larger than 100 bp) discovered and validated on HuRef genome [3] in comparison to NCBI Human Genome (hg18) from chromosome 18, 19 and 20 (respectively 109, 121 and 62 deletions), and imposed them on hg18 genome (by removing the same segments from the Hg18 genome). In next step we produced short paired-end reads exactly similar to what Illumina machine would have produced with a normal distribution of fragment size which spans from 172 to 242 bp long form the altered Hg18 (imposed deletions). The reads produced are mapped back to the hg18 genome, using mrsFAST algorithm [4]. We run our original VariationHunter and the new VariationHunter-CR on the discordant paired-end read alignments. The results indicate that even in this very simple simulation (where these are simply very low possibilities of having conflicting valid clusters) the VariationHunter based on conflict resolution (VariationHunter-CR) has lower False positive rate, while having the same true positive rate (in chr18 the true positive rate is also quite higher in VariationHunter-CR) in comparison to original VariationHunter [5]).

The results of this simple simulation is shown in the Table 1.

		VariationHunter [5]			VariationHunter-CR		
chromosome	Deletions From HuRef Imposed on HG18	Predicted	True Positive	False Positive	Predicted	True Positive	False Positive
chr18	109	97	**79**	**18**(**18.5**%)	96	**80**	**16**(**16.6**%)
chr19	121	114	92	**22**(**19.3**%)	111	92	**19**(**17.1**%)
chr20	62	55	43	**12**(**21.8**%)	53	43	**10**(**18.8**%)

Table 1: In this table we show results of a simple simulation we have run on Hg18, using large deletions found on HuRef genome on chromosome 18, 19 and 20. These are results predicted using original VariationHunter[5] and VariationHunter-CR. As it can been seen, even in this simple control simulation/experiment, the second method has less false positive discovery having the same number of true positive discovery.

References

[1] Papadimitriou CH, Yannakakis M (1988) Optimization, approximation, and complexity classes (extended abstract). pp. 229–234.

[2] Halldrsson M (1993) Approximating the minimum maximal independence number. Inform Process Lett 46:169–172.

[3] Levy S, Sutton G, Ng PC, Feuk L, Halpern AL, et al. (2007) The diploid genome sequence of an individual human. PLoS Biol 5:e254.

[4] Hach F, Hormozdiari F, Alkan C, Birol I, Eichler EE, et al. (2010) Cache oblivious algorithms for high throughput read mapping. unpublished .

[5] Hormozdiari F, Alkan C, Eichler EE, Sahinalp SC (2009) Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes. Recomb 2009/Genome Research 19:1270–1278.