DeepSV: a deep learning based variant scorer

Félix Raimundo
Department of Computer Science
École Polytechnique
Palaiseau, France
felix.raimundo@inria.fr

February 15, 2018

Abstract

In this paper, we present DeepSV, a deep learning based method for scoring the likelihood that a set of adherent read pairs are caused by a structural variant, by opposition to mapping or sequencing errors.

We identified a region of fixed size that contains enough signal to make classification with deep learning methods, and thus allowing us to move away from the previous methods based on linear classifiers applied on few hand-crafted features.

1 Definitions

In this papers we will be using the sequencing terms as defined in [1], whose most relevant ones are copied here.

Segment A contiguous sequence or subsequence.

Read A raw sequence of DNA that comes off a sequencing machine.

Coverage Number of reads mapped at a specific location in the reference genome.

Read Pair Long sequence of DNA of which the sequencing machine reads the two extremities. Each of the two extremities are reads and are associated a direction and are separated by a distance called insert size.

Note: For the duration of this paper, we will assume that the two reads of a read pair are both directed to the center, i.e. the first read is directed from left to right, called forward, and the second right to left, called backward.

2 Properties of the data

In this paper we will be making the following two assumptions:

Property 1 The distance between two reads of a read pair, called insert size, follows a normal distribution.

Property 2 The length of reads, called read size, follows a normal distribution.

Note: there are no guarantees that they are true, nevertheless they are a good approximation of reality and are used in most other variant callers (see [2], [3], [4], [5], [6]).

Note: The distance between two reads of a pair generated from a target genome can be different once the two reads are mapped on a reference genome, the analysis of that phenomenon is the basis of our study.

Hypothesis 1 : During the rest of this document we will assume that the insert size and read size are constant when the reads are generated, indeed the two distribution have a low standard deviation, this approximation is thus sensible. All the future claims stay valid with this approximation, but would make the text much less readable.

Section *Dealing with reality* will describe how we deal with real data, where these assumptions are not correct.

Property 3 Because of properties 1 and 2, and hypothesis 1, it follows that the second pair of a read pair is fully contained within $[insert\_size, insert\_size + read\_size]$ of the end of the first one in the genome from which the reads are extracted.

2.1 Structural Variants

Variant We call variant, a difference between the genome of the person being sequenced and the reference genome, they are the cause of the genetic diversity in the population.

Some variants are called *structural variants*, because they are considered so large that they come from a structural difference. The limit between regular variant and structural variant does not have a proper definition, but they are usually considered to be changes affecting more than 50bp.

Structural variants are separated in two categories:

- Balanced rearrangements: which do not affect the total number of nucleotides in the genome.
- Imbalanced rearrangements: which affect the total number of nucleotides in the genome, also called Copy Number Variation (CNV).

The CNVs are themselves divided in three categories:

- Deletions: where a large portion of the genome is deleted relative to the reference.
- Tandem duplication: where a large portion of the genome is duplicated, potentially multiple times; the duplicates are directly next to each other.
- Interspersed duplication: where a large portion of the genome is duplicated, potentially multiple times; the duplicates are separated by segments of DNA.

Likewise the balanced rearrangements are divided in two categories:

- Translocation: where a large portion of DNA is moved to another location in the genome (potentially on another chromosome).
- Inversion: where a large portion of DNA has the order of its nucleotides reversed.

In our study we will only work on deletions, tandem duplications, and inversions. We however believe that our method could be extended to work on the other types of variants (see *Future Work*).

Breakpoint We call breakpoints the locations where structural rearrangements happen in the genome , there are:

- 1 in the case of deletions.
- n for tandem duplications with n duplicates.
- n for interspersed duplications with n duplicates.
- 2 for inversions
- 2 for translocations.

2.2 Methods of detection

The current state of the art algorithms for finding structural variants are based on a mixture of the four following methods (see [7] or [8] for a deeper review):

- Assembly based (AS), does *de novo* assembly on regions suspected to contain an SV. They are rarely used as: assembly is computationally expensive, and not always possible in regions with repetitive sequences.
- Read count (RC), observes the distribution of coverage in the genome. It can only detect CNVs as reads in translocations and inverted regions are properly mapped. Furthermore reads are not generated uniformly in the genome from which the reads come, it is thus not an optimal method alone.

- Split reads (SR), uses reads that fall on the breakpoint. Those reads are unmapped, as part of them is in the rearrangement. These reads are split so that they can be mapped properly. As the split reads are smaller than regular reads they cannot always be mapped unambiguously.
- Paired reads (PR), uses pairs of mapped reads that have an insert size far away from the mean insert size, or who do not have the expected orientation. This method is often used to identify regions that contain SV and followed by a finer analysis of the candidate.

In the case of DeepSV we use paired reads to identify regions of interest and use a pattern recognition model on the selected regions to decide whether they contain an SV. We expect our model to use RC and PR methods for predictions. AS and SR methods can be used to locate the breakpoints with higher precision once the prediction is made.

The next sections will explain what patterns we expect to find in these regions.

2.3 Anomalous paired reads

Anomalous read pairs (ARP) are read pairs that do not have the normal orientation or distance.

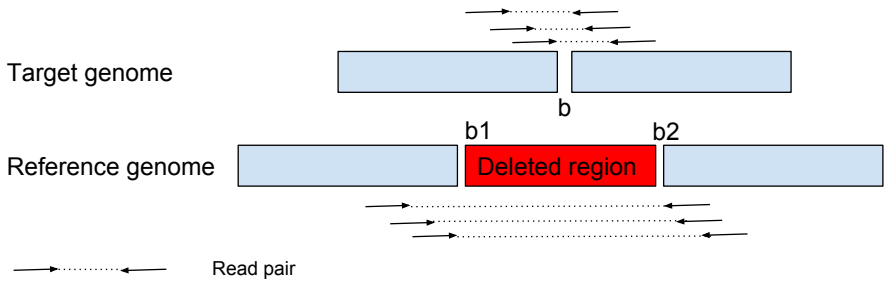
Property 4 Assuming no errors in the reads or mapping, ARPs contain a breakpoint between their two reads.

Indeed, if one read was on the breakpoint, it could not be mapped as a part of it would be on a large rearrangement.

Respectively, if both reads were the same side of a breakpoint, they would be mapped normally and would thus not be caught as anomalous.

Property 5 Because of properties 3 and 4 we can conclude that all reads in ARPs will be at most $read\_size + insert\_size$ away from the breakpoint in the genome that generated the reads.

2.3.1 Deletion



In the case of deletions there is a single breakpoint b, the deleted region is $[b1, b2]$.

Assuming no errors in the reads or mapping:

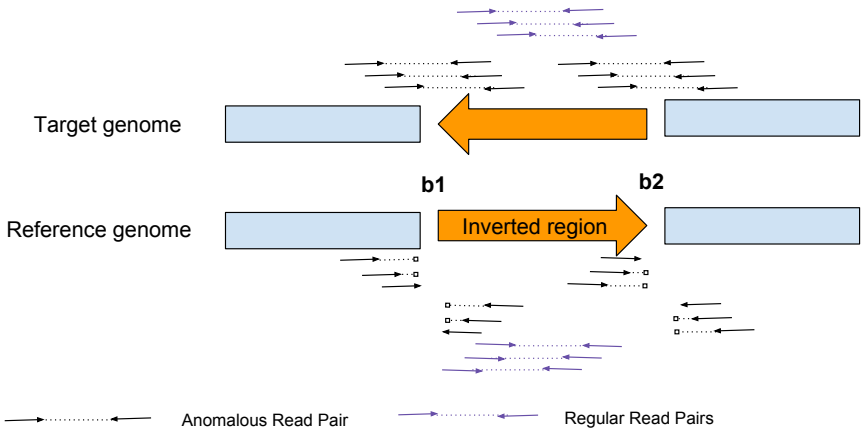
- reads to the left of b are mapped left of b1 in the reference genome.
- reads to the right of b are mapped to the right of b2 in the reference genome.
- no reads should be mapped in $[b1, b2]$.

We can conclude that given a set of ARPs supporting a deletion, b1 will be contained within $insert\_size + read\_size$ of the leftmost read, and b2 will be contained within $insert\_size + read\_size$ of the rightmost read.

Furthermore, the previous observations, allows us to tell that all ARPs will be contained in that region. If we also add some content from the deleted section we can observe the coverage there and compare it to the coverage outside the deleted region.

We have thus constructed two intervals containing all the ARPs and RC information, these two intervals' sizes are only dependant on the insert size and read size, not the size of the deletion.

2.3.2 Inversion



Here ARPs are pairs where one read is in the inversed region and the other is in the regular one (i.e. either one read left of b1 or one read right of b2). Indeed reads between b1 and b2 will be mapped properly (because BWA is not sensitive to the order of the reads).

Reads in the inversed region will be mapped in the wrong direction, and farther away than expected (as the region is inversed they will be moved closer to the other breakpoint).

We can see that if one read is left of b1, the other one will be mapped close to b2 inside the inversed region and in the wrong direction, and that if one read

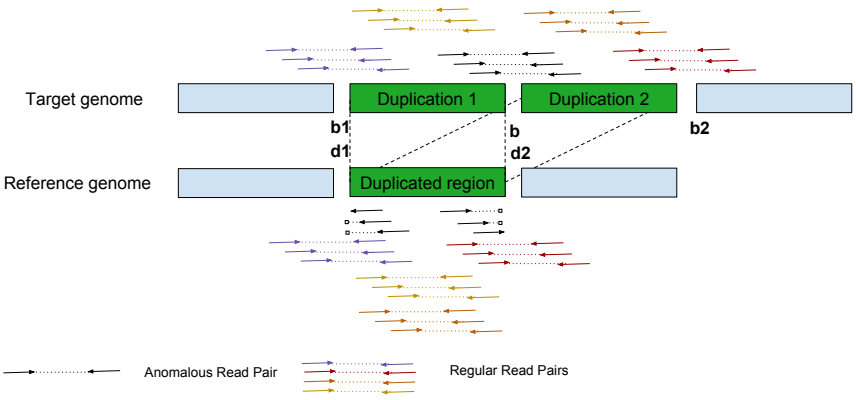
is to the right of b2, the other one will be mapped close to b1 in the inversed region and in the wrong direction too.

We can also see by using properties 1 and 2, that reads outside the inversed region will be at most `insert_size + read_size` away from the breakpoints and that the same is true for reads inside the inversed region.

Lemma 4 ARPs supporting an inversion will be distributed around the two breakpoints within `2*(insert_size + read_size)`.

Lemma 3 Respectively, should we have a group of ARPs supporting the same inversion we can say with a known confidence interval, that most ARPs around the left breakpoint will be contained within `2 * (insert_size + read_size)` of the beginning of the leftmost supporting read. The same is trivially true for the right breakpoint.

2.3.3 Tandem Duplication



Here ARPs, are pairs that have one read on each side of breakpoint b, indeed read pairs fully contained between b1 and b or b and b2, will be mapped properly on the duplicated region.

Lemma 6 ARPs are aligned in the wrong direction (`<- ->` instead of `-> <-` for example), and inside the duplicated region, one within `insert_size + read_size` to the right of d1, the other within `insert_size + read_size` left of d2. The read count in the duplicated region will be about the double of outside the duplicated region.

Lemma 7 It follows that should a region be duplicated more than once, the ARPs will be distributed the same way as with tandem duplication (except for higher read count and more ARPs).

2.4 Conclusion

Theorem 1 Putting all the lemmas together we can conclude that given a set of structural variants, there are two windows of size `2 * (insert_size + read_size)` each, centered on the two breakpoints in the reference genome that contains most of the ARPs that will support it.

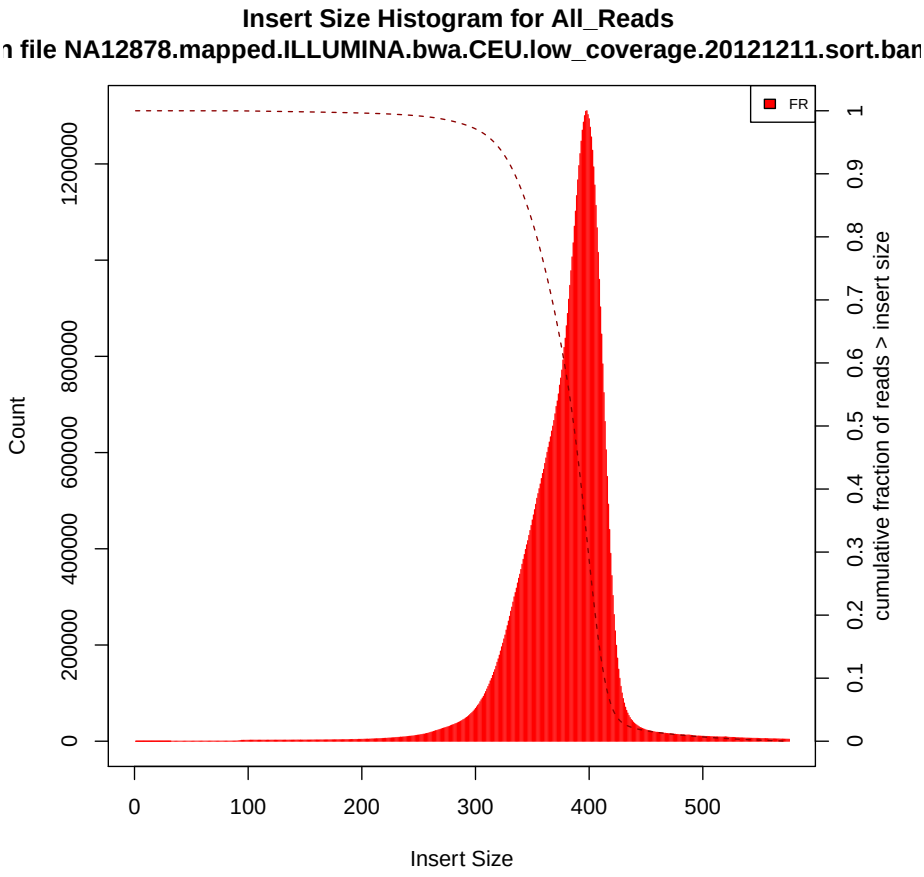
Theorem 2 Given a type of variant and a set of supporting ARPs, we can construct two windows of size `2 * (insert_size + read_size)` that will contain the breakpoints in the reference genome.

Note For tandem CNVs (and deletions) windows of size `insert_size + read_size` would have sufficed to contain all the ARPs, however they would not have contained reads outside of the duplicated area (inside the deleted area), which would make us lose the RC information. Taking larger windows also allows us to have a single window size for all our variants, and thus using the same model for all the variants.

Note All the window sizes given here assumed that the reads and inserts were of the same size, when we know that they are variable, adding a few standards deviations will allow us to catch the read pairs with higher insert size or read size. [**?**]

3 Dealing with real data

As said before, the empirical insert size does not follow a normal distribution. We can see that it is skewed towards lower values. However, we can see that the distribution is well centered on its mean: 400 bp.



Listing 1: Picard command

```
#!/bin/bash
java -jar picard.jar CollectInsertSizeMetrics \
    I=input.bam \
    O=insert_size_metrics.txt \
    H=insert_size_histogram.pdf \
    M=0.5
```

We are aware of two reasons that could cause such a skew:

- Mobile element insertions of ALUs.
- Poor mapping in ambiguous regions.

Picard tells us that a width of 165 centered on the median, 386bp, will catch 95% of the reads, which will be considered a sufficient approximation. In consequence when building our window we will consider an insert size of 467bp.

References

References

[1] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, and Richard Durbin. The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, August 2009.

[2] Tobias Rausch, Thomas Zichner, Andreas Schlattl, Adrian M. Stütz, Vladimir Benes, and Jan O. Korbel. DELLY: structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics (Oxford, England)*, 28(18):i333–i339, September 2012.

[3] Kai Ye, Marcel H. Schulz, Quan Long, Rolf Apweiler, and Zemin Ning. Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics (Oxford, England)*, 25(21):2865–2871, November 2009.

[4] Xiaoyu Chen, Ole Schulz-Trieglaff, Richard Shaw, Bret Barnes, Felix Schlesinger, Morten Källberg, Anthony J. Cox, Semyon Kruglyak, and Christopher T. Saunders. Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. *Bioinformatics*, 32(8):1220–1222, April 2016.

[5] Robert E. Handsaker, Vanessa Van Doren, Jennifer R. Berman, Giulio Genovese, Seva Kashin, Linda M. Boettger, and Steven A. McCarroll. Large multi-allelic copy number variations in humans. *Nature genetics*, 47(3):296–303, March 2015.

[6] 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. *Nature methods*, 6(9):677–681, September 2009.

[7] Ke Lin, Sandra Smit, Guusje Bonnema, Gabino Sanchez-Perez, and Dick de Ridder. Making the difference: integrating structural variation detection tools. *Briefings in Bioinformatics*, 16(5):852–864, September 2015.

[8] Peter H. Sudmant, Tobias Rausch, Eugene J. Gardner, Robert E. Handsaker, Alexej Abyzov, John Huddleston, Yan Zhang, Kai Ye, Goo Jun, Markus Hsi-Yang Fritz, Miriam K. Konkel, Ankit Malhotra, Adrian M. Stütz, Xinghua Shi, Francesco Paolo Casale, Jieming Chen, Fereydoun Hormozdiari, Gargi Dayama, Ken Chen, Maika Malig, Mark J. P. Chaisson, Klaudia Walter, Sascha Meiers, Seva Kashin, Erik Garrison, Adam Auton, Hugo Y. K. Lam, Xinmeng Jasmine Mu, Can Alkan, Danny Antaki, Taejeong Bae, Eliza

Cerveira, Peter Chines, Zechen Chong, Laura Clarke, Elif Dal, Li Ding, Sarah Emery, Xian Fan, Madhusudan Gujral, Fatma Kahveci, Jeffrey M. Kidd, Yu Kong, Eric-Wubbo Lameijer, Shane McCarthy, Paul Flicek, Richard A. Gibbs, Gabor Marth, Christopher E. Mason, Androniki Menelaou, Donna M. Muzny, Bradley J. Nelson, Amina Noor, Nicholas F. Parrish, Matthew Pendleton, Andrew Quitadamo, Benjamin Raeder, Eric E. Schadt, Mallory Romanovitch, Andreas Schlattl, Robert Sebra, Andrey A. Shabalin, Andreas Untergasser, Jerilyn A. Walker, Min Wang, Fuli Yu, Chengsheng Zhang, Jing Zhang, Xiangqun Zheng-Bradley, Wanding Zhou, Thomas Zichner, Jonathan Sebat, Mark A. Batzer, Steven A. McCarroll, 1000 Genomes Project Consortium, Ryan E. Mills, Mark B. Gerstein, Ali Bashir, Oliver Stegle, Scott E. Devine, Charles Lee, Evan E. Eichler, and Jan O. Korbel. An integrated map of structural variation in 2,504 human genomes. *Nature*, 526(7571):75–81, October 2015.