An Nguyen

PID: A13206547

Ledell Family

Somatic Structural Variation Detection in Human Genomes with Machine Learning

Abstract

Structural variants (SVs) have been attributed to a number of genetic disorders due to having large effects on DNA sequences (Freeman et al., 2006). SVs, although less frequent than other mutations, take up more of the genome than other classes of mutations (Sudmant et al., 2015). As such correctly identifying SVs and characterizing becomes an important objective. Next generation sequencing (NGS) can be useful in detecting SVs but have been attributed to be limited by generating many false positive detections of SVs due to coverage variation and alignment errors. While traditional methods such as with array comparative genomic hybridization (aCGH) and Multiplex Ligation-dependent Probe Amplification (MLPA) have limitations in accuracy and processing rate respectively (Mu et al., 2019). It is difficult to characterize these somatic SVs without paired tissues for comparative analysis. Somatic SVs essentially have a lot of noise but very little signal of detection. Complications in detection, indicates that machine learning can be a powerful tool to find somatic SVs after adequate training with a decrease rate of false positive SVs than traditional NGS. We summarize our progress in building a machine learning model using an algorithm training set for the model to learn and predict somatic variations.

Introduction

The method for detecting and classifying SVs in NGS can be summarize with four features: read-pair(RP), split-read(SR), read-depth(RD) and k-mers. RP considers using a standard reference genome to see disturbances in genome when mapping distances between two paired-end reads indicating most kind of structural variants. However, mapping distances/insert size follows a standard distribution therefore this method cannot identify small or large variations due to being outliers. The repetitive nature of sequences in human genome can also cause incorrect mapping leading to false positives. SVs will also leave SR signals near the break point of variants and by breaking short reads into fragments and mapping them separately to the reference genome is possible for the detection of insertions, deletions, and inversion by looking at how the breakpoints align in relations to each other. Large and repetitive sequences cause limitations to this method. RD method involve looking at the proportions of number of copies in the reference genome comparatively with duplicated and deleted regions to have higher or lower coverage. By looking at the RD or coverage it is possible for detecting larger duplicated and deleted regions but unable to identify inversion and less effective against repetitive regions due to lower sensitivity, a single loss no longer causes a noticeable change in the proportion of copies (K & G, 2015). K-mers is a relatively new feature used in detecting SVs and in essentially it is a word frequency method. Similar sequences share similar words (k-mers) and through math operations it is possible to get a good measure of their similarities. Smaller k-mers are used when

sequences are very much different and longer k-mers denote similar sequences (Zielezinski, Vinga, Almeida, & Karlowski, 2017).

Common type of SVs includes deletions (DELs) , insertions (INSs) , duplications (DUPs), and inversions (INVs) that accounts for the majority of variation in nucleotides in the genome.(Sudmant 2015) Inversions leave no net copy change in the genome while others causes removal or addition of new genetic materials also known as copy number variations (CNVs). Duplications can happen between adjacent nucleic acid or over large sequences depending on the distance between the copies. Insertions are based on the type of sequence, which includes mobile element insertions (MEIs), nuclear insertions of mitochondrial genome (NUMTs), viral element insertions (VEIs) and other sequences (Kosugi et al., 2019).

Somatic variations occur due to changes in single nucleotide polymorphism (SNPs), insertions, and deletions otherwise known as INDELs with some structural variations having have very large size about 1Mb. These mutations occur after fertilization and are not inherited. From past SVs data we hypothesize that most somatic structural variants are smaller than 1Mb and remains difficult to be accurately identify by NGS resulting in high false positive rate, instead using a machine learning model can help decrease false positive discovery rate (FDR).

Methodology:

Our first objective consists of creating a robust training set to train our machine learning model. We used data from the genome in a bottle (GIAB) consortium Ashkenazi father-mother-child trio ("GIAB," n.d.). Upon further inspection of the data it was not in the correct format that was needed for proper analysis and set us back in our original plan. Therefore, we decided to make alignments from raw reads instead of using the mapped files from GIAB. These steps require downloading many FASTQ files, text-based files storing the nucleotide sequences from sequencing experiments, available on GIAB for the 3 family members, aligning the correct pairs of FASTQ, comparing them to our reference genome, merging them through refining steps into the proper merged BAM files, a binary version of the alignment data, to be used (Regier et al., 2018). This process consisted of me running multiple processes in the form of jobs on the Triton Shared Computing Cluster (TSCC), UC San Diego high performing computing system.

I learned a great deal of computing using bash, a shell language, for the linux operating system as this is how I run jobs to do numerous amounts of task that requires high computing power on TSCC. Downloading the FASTQ comes with validating and verifying that each individual file have been properly downloaded with no error, this proves to be a lengthy task as there was hundreds of FASTQ to download. Once completed my next step consist of running an alignment script utilizing Burrows-Wheeler short read alignment tool (BWA). This method has its advantages in being fast and efficient while achieving similar accuracy with traditional method and is consider to be the new standard for creating BAM files (Li & Durbin, 2009).

Once this was completed the bam files were sorted and duplicate reads were marked, realigned, recalibrated, and finally the reads for files were ready. The step after this consists of merging the bam files to get a deep coverage bam file (300x) and a high coverage (50x) bam files for father-mother-son. I ran samtools flagstat, a tool developed by illumina called index

depth, afterwards to count and verify the amount of coverage within the files and ended up getting only around 200x for deep coverage and 50x for high coverage yielding lower than expected (*Illumina/paragraph*, 2017/2019) . This result was further verified by running bedtools genomecov ("BEDTools: a flexible suite of utilities for comparing genomic features | Bioinformatics | Oxford Academic," n.d.) and writing a script to calculate each of the bam files finding the mean coverage reported by bedtools which yielded a higher amount of coverage than what was gathered by running samtools flagstat closer to 300x for deep coverage.

Utilizing the coverage data, we ran "mixing experiment" of the mom and dad on strictly chromosome 19 using deep coverage data at varying levels to mimic mosaicism, new mutations that occur after fertilization in some cells but not all. Conceptually if a mutation occurs after the first cell division after fertilization, then approximately 50% of the cells in an adult will carry that mutation onwards. Similarly, if a mutation occurred after the second cell division, 25% of the cells in that adult will have that mutation compared to first cell. This experiment replicates that concept but can look at the variation changes in each cell using coverage data by comparing one cell with another cell, the choice doesn't matter in this case. We picked the mom (spike in) to be compared with the dad (background). We start with a 1:1 dad to mom ratio capping at 200x coverage meaning that both dad and mom need 100x coverage each. However, because both have higher coverage than 100x and each need to be downsized using simple ratios. The next ratio is 3:1 with dad at 150x and mom 50x, follow by 7:1 with dad at 175x and mom at 25x up to 63:1 with each ratio's coverage downsized by the deep coverage data for dad and mom. A range of allele fractions is determined dependent on the mother's genotype of the SV. The table generated here is useful to make mixed subsets of the coverage data for our experiment. I ran scripts with the subsets to determine the breakpoints.

To verify that the data we made from GIAB is the same as the GIAB Illumina call set I wrote a script to filtered out SV lengths and look at only the deletions or DELs. This gives us an idea of what SVs overlapped and don't overlap with each other. Additionally, I wrote a script to go through the GIAB Illumina call set and count the number of SV size based on the SV size range and genotype of the dad and mom. A distribution of INTs and DELs was obtained by from the call set as well.

Results

Chrom 19 (MOM+DAD)	Allele Fraction Range(200x)	REF(DAD)	SPIKE(MOM)	Calculation DAD	Calculation Mom		
	1/2 - 1/4	100	100	0.378357927	0.347463516		DAD Deepcov at Chrom 19
	1/4 - 1/8	150	50	0.56753689	0.173731758		264.3
	1/8 - 1/16	175	25	0.662126372	0.086865879		
	1/16 - 1/32	187.5	12.5	0.709421112	0.04343294		MOM Deepcov at Chrom 19
	1/32 - 1/64	193.75	6.25	0.733068483	0.02171647		287.8
	1/64 - 1/128	196.875	3.125	0.744892168	0.010858235		

Figure 1: Data obtained before the mixing experiment. Calculations are simple ratios that equates the (REF/Spike) / x = (Dad/Mom Deepcov) / 1. These calculations are useful for creating mixed subsets using the deep coverage data.

	Sample Name	Chrom	Depth	Sample Name	Chrom	Depth
1	HG003.100subs.chr19.genomecov	19	99.97900415	HG004.100subs.chr19.genomecov	19	99.98658056
2	HG003.150subs.chr19.genomecov	19	149.9609218	HG004.50subs.chr19.genomecov	19	50.00145742401827
3	HG003.175subs.chr19.genomecov	19	174.9754564	HG004.25subs.chr19.genomecov	19	25.01501952
4	HG003.187.5subs.chr19.genomecov	19	187.4760377	HG004.12.5subs.chr19.genomecov	19	12.48751616
5	HG003.193.75subs.chr19.genomecov	19	193.7391827	HG004.6.25subs.chr19.genomecov	19	6.243002911
6	HG003.196.875subs.chr19.genomecov	19	196.8638772	HG004.3.125subs.chr19.genomecov	19	3.138115736

Figure 2: Verification that the creation of the subset coverage went well. The depth reads matches the coverage sample number.

DAD GT	MOM GT	NUMBER of SVs	NUMBER of SV SIZE BIN	[DATA] NUMBER of SV SIZE BIN
0/1	0/0	455	<50	0
1/1	0/0	181	51-100	2
0/0	0/0	6	101-250	4
0/0	0/1	471	251-500	1908
0/1	0/1	557	501-1kb	584
1/1	1/1	1163	1kb-5kb	937
			5kb-10kb	214
			10kb-100kb	60
			>100kb	1
			Total	3710

Figure 3: Table shows the distribution of SV size bin over different ranges. As shown here most SVs are under 1MB with majority of SVs between 250-10kb

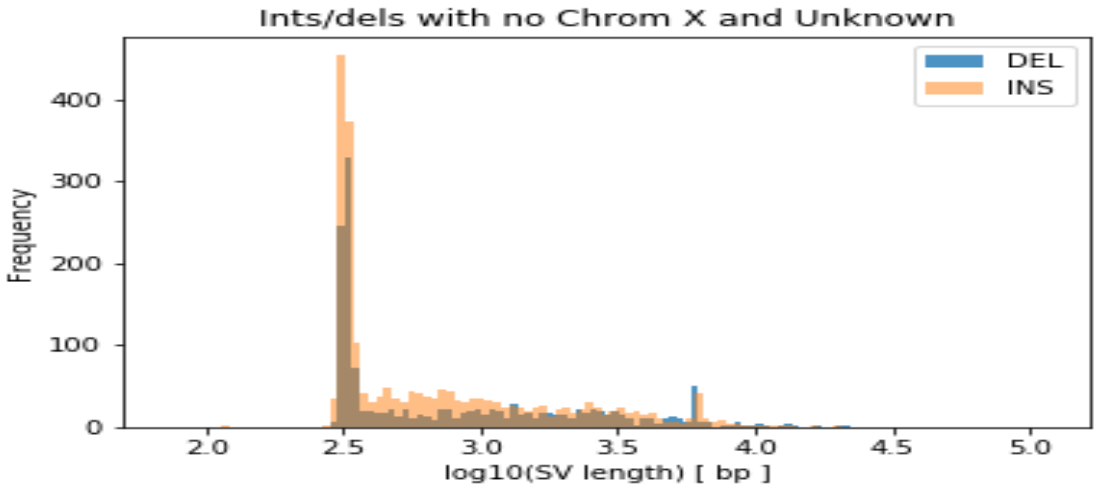

Figure 4: A graph showing the distribution of insertions and deletions within the GIAB Illumina call set.

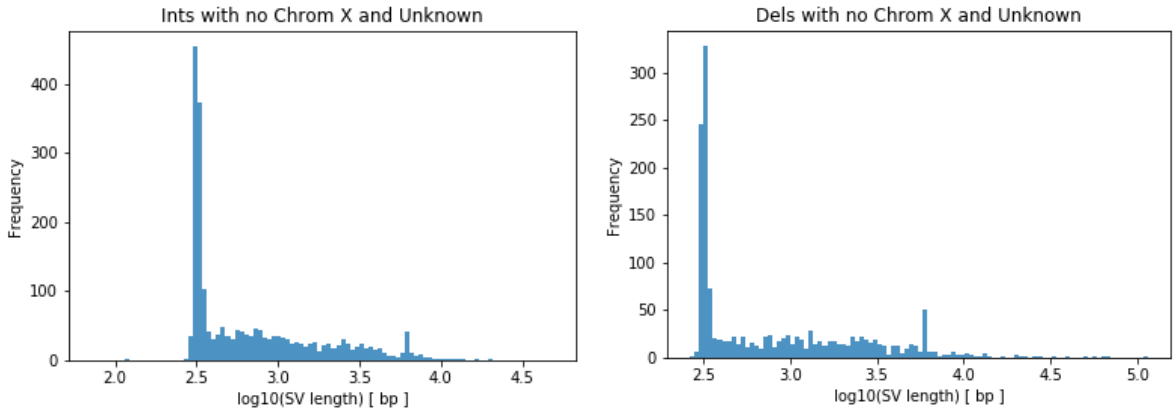

Figure 5: graphs that individually shows insertions and deletions distributions. The gap in beginning of insertions indicates that there are smaller size insertions as compared to deletions

Discussion

The creation of the training set data from the GIAB consortium Ashkenazi father-mother-child trio proves to be provide high quality data for our mixing experiments but took much longer than intended. The limitation in our pre-work creating the training set is that it is only 1 training set on chromosome 19 and we would need more examples in order to train our machine learning model. Analyzing the GIAB Illumina call data is very helpful in letting us know that our training set is good with very few false positives. It is important to make sure that our training set and data is robust because training our model with inadequate data will not allow our model to discover SVs accurately. Future steps will be to expand our training set and optimize how to train our model. We will then apply the model to call somatic SVs in the sperm of 8 fathers, who have had a least one child with autism which had been determined in our preliminary work. We will then validate the data using ddPCR and amplicon sequencing to see the percentage of false discovery rate.

Somatic SVs have been implicated in numerous diseases and are likely to be the driving force around functional changes (Freeman et al.2006) In fact spontaneous SVs in father's germline has been correlated with disorders such as autism (Brandler et al., 2016) Furthermore somatic mutations that arises during prenatal development can cause many neurological disorders and have been linked to cancer (Poduri, Evrony, Cai, & Walsh, 2013). As a result, proper detection and characterization of these Somatic SVs can give us more information on what their function is in driving these diseases. Although NGS and novel technologies have successes in identifying somatic SVs, there is still high false positive rates (Mu et al., 2019). As a result, many methods have to be used in order to accurately predict SVs. The advent of machine learning can possibly help solve this solution and find better predict SVs within a convoluted system through adequate training.

Appendix

The purpose of the project is to help our lab develop a machine learning model to detect somatic structural variations (SVs). SVs are widely associated with neurological diseases and cancers and accurately detecting them can help our understanding of their role. More particularly my job is to help pave the foundation by working to create the training set needed to train the machine learning model. The benefit of this project is that our machine learning model can be a useful tool in the future for our lab to use in identifying somatic SVs and give the scientific community more insight on the successes and limitations of using a machine learning model approach towards detection of somatic SVs. I spent majority of my time learning to navigate the Triton Shared Computing Cluster (TSCC), UC San Diego high performing computing system, using bash on linux operating system to be able to work with the data in Genome in a Bottle (GIAB) consortium, writing scripts with python to parse and generate tables and figures from data for our training set. I learned a lot about bioinformatics and computing this summer which improves my own understanding of doing modern day research as it has begin incorporating bioinformatics in addition to wet lab. Additionally, project have given me a taste of what it like to work independently and to find answers in a field that I have little experience in. I plan to continue helping the lab with the project.

The funds through the undergraduate research scholarship program given to me by the Ledell family was for travel, parking, food, living expenses, and insurance payment. I am thankful that with their generosity I was able to experience being paid to do research at the Gleeson Lab.

The average amount of hours I spent doing research a week is around 35 hours.

References

BEDTools: a flexible suite of utilities for comparing genomic features | Bioinformatics | Oxford Academic. (n.d.). Retrieved September 5, 2019, from https://academic.oup.com/bioinformatics/article/26/6/841/244688

Brandler, W. M., Antaki, D., Gujral, M., Noor, A., Rosanio, G., Chapman, T. R., … Sebat, J. (2016). Frequency and Complexity of De Novo Structural Mutation in Autism. *American Journal of Human Genetics*, *98*(4), 667–679. https://doi.org/10.1016/j.ajhg.2016.02.018

Freeman, J. L., Perry, G. H., Feuk, L., Redon, R., McCarroll, S. A., Altshuler, D. M., … Lee, C. (2006). Copy number variation: New insights in genome diversity. *Genome Research*, *16*(8), 949–961. https://doi.org/10.1101/gr.3677206

GIAB. (n.d.). Retrieved September 2, 2019, from The Joint Initiative for Metrology in Biology website: https://jimb.stanford.edu/giab

Illumina/paragraph [C++]. (2019). Retrieved from https://github.com/Illumina/paragraph (Original work published 2017)

K, Y., & G, H. (2015). Structural Variation Detection from Next Generation Sequencing. *Journal of Next Generation Sequencing & Applications*, *01*(S1). https://doi.org/10.4172/2469-9853.S1-007

Kosugi, S., Momozawa, Y., Liu, X., Terao, C., Kubo, M., & Kamatani, Y. (2019). Comprehensive evaluation of structural variation detection algorithms for whole genome sequencing. *Genome Biology*, *20*(1). https://doi.org/10.1186/s13059-019-1720-5

Li, H., & Durbin, R. (2009). Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics*, *25*(14), 1754–1760. https://doi.org/10.1093/bioinformatics/btp324

Mu, W., Li, B., Wu, S., Chen, J., Sain, D., Xu, D., … Lu, H.-M. (2019). Detection of structural variation using target captured next-generation sequencing data for genetic diagnostic testing. *Genetics in Medicine*, *21*(7), 1603–1610. https://doi.org/10.1038/s41436-018-0397-6

Poduri, A., Evrony, G. D., Cai, X., & Walsh, C. A. (2013). Somatic Mutation, Genomic Variation, and Neurological Disease. *Science (New York, N.Y.)*, *341*(6141), 1237758. https://doi.org/10.1126/science.1237758

Regier, A. A., Farjoun, Y., Larson, D. E., Krasheninina, O., Kang, H. M., Howrigan, D. P., … Hall, I. M. (2018). Functional equivalence of genome sequencing analysis pipelines enables harmonized variant calling across human genetics projects. *Nature Communications*, *9*(1). https://doi.org/10.1038/s41467-018-06159-4

Zielezinski, A., Vinga, S., Almeida, J., & Karlowski, W. M. (2017). Alignment-free sequence comparison: Benefits, applications, and tools. *Genome Biology*, *18*(1). https://doi.org/10.1186/s13059-017-1319-7