Structure Variation Summarization

Motivation

1: Design and implement a pipeline that can be used for Structure variation detection by using Pacbio HiFi reads (Sequel II reads).

2: Our requirements for this pipeline

1. Only for Pacbio Sequel II reads
2. Only for human genome data
3. High Accuracy
4. High Sensitivity (recall)
5. Link with LIMS for result management
6. Fully automatic
7. Can handle some requirement in the future
   1. Combine the result from SMRT Link directly
   2. Parameter adjustment based on different set of sequel II reads
      1. Regular
      2. Low coverage
      3. Ultra-low coverage

Strategy

1. Check which SV tools should be used
   1. Know their performance
2. Check how to organize our inhouse pipeline
   1. Check the existing pipeline
   2. Check the steps
      1. Data preprocessing
      2. Quality control
      3. Mapping
      4. Call SV
      5. Result merging
      6. Data visualization
      7. Release final report
      8. Result verification (for building benchmark)
3. Data Set
   1. Check which public data set can be used as benchmark
   2. Check which tools that can be used to generate the simulating data.
      1. For the preparation of benchmark testing

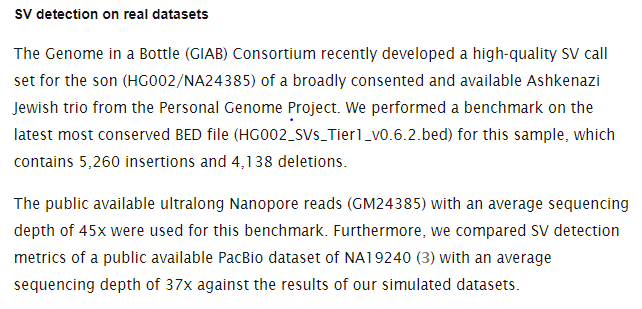
Survey

**1: Which tools can be used for mapping**

**2: Which tools can be used for SV detection**

1. Sniffles (ref 1)

**3: Existing public real dataset**

1. 

**4: Simulator**

1. <https://medium.com/computational-biology/a-simple-introduction-to-read-simulators-bbeff4f0c0c6>
   1. PacBio Simulators (presented)
      1. PBSIM -> provide two different way to generate simulated reads
         1. Model-based simulation
         2. Sampling-based simulation
      2. LongISLND
      3. SimLoRD (presented)
         1. <https://bitbucket.org/genomeinformatics/simlord/src/master/simlord/>
            1. Python code
         2. Is a TGS read simulator based on the Pacific Biosciences SMRT error model.
            1. Simulate fixed-length reads by providing **the number of reads**
            2. Simulate fixed-length reads by providing **the coverage**
      4. NPBSS
      5. PaSS
   2. ONT Simulators
      1. NanoSim
         1. https://academic.oup.com/gigascience/article/6/4/gix010/3051934
      2. Nanopore SimulatION
      3. DeepSimulator
      4. DeepSimulator1.5

**5: Result verification**

**6: Existing pipeline for SV detection by using long reads (Pacbio or ONT)**

**1. Structural variant calling - long read data**

1. The university of Melbourne

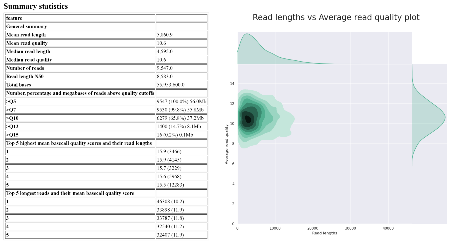
<https://www.melbournebioinformatics.org.au/tutorials/tutorials/longread_sv_calling/longread_sv_calling/>

* 1. Data set used
     1. bacterial dataset (nanopore)
     2. human clinical dataset (pacbio)

1. SV calling pipeline
   1. Three key phases include
      1. data exploration
      2. calling variants
      3. then interpreting our findings
   2. The pipeline consists of 5 key steps
      1. Read QC
         1. NanoPlot.
            1. Want to get a summary of our read set
            2. creates plots to summarise the length distribution and quality of our read set.
            3. NanoPack:

https://www.pacb.com/publications/nanopack-visualizing-and-processing-long-read-sequencing-data/

Can be used for both ONT and PacBio long reads



* + - 1. Filtlong
         1. be used to remove short reads, or those with patches of low quality

As our SV caller, sniffles, looks at alignment quality when identifying structural variation, it is important we remove reads with low-quality patches.

* + - * 1. will reduce erroneous read mapping and will improve our results.
        2. Default parameters

Min. Length 1000

remove any reads which are less than 1000 bp

Min. window quality 9

those where a section (250bp) of the read has mean quality below Q9

* + 1. Alignment
       1. Minimap2
          1. pass the output alignment BAM file to our SV caller.
    2. SV Calling
       1. Sniffles
          1. output variant calls in VCF format,
          2. requires an alignment file when calling variants

split reads

soft clipping of reads

* + - * 1. Set general options

Minimum Support: 10 (default) or 5 (adjusted)

* + - * 1. Requires alignments to contain the ‘**MD tag’** in our BAM file

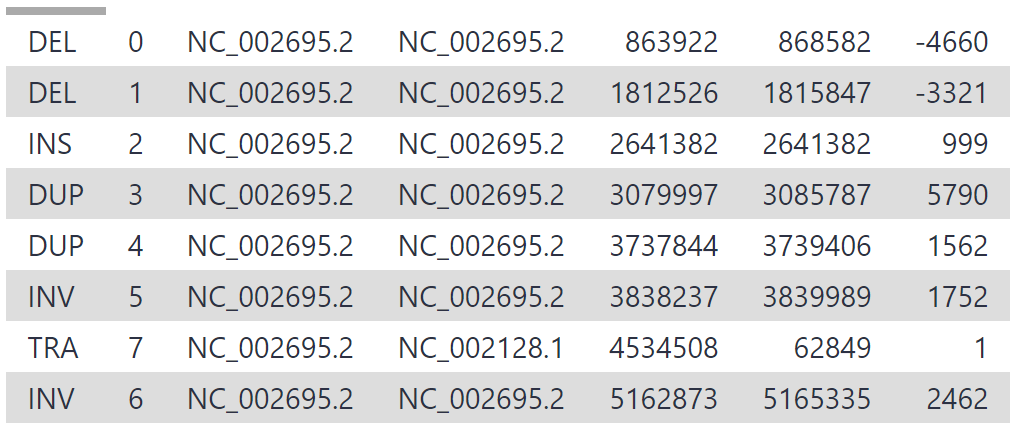
This is a condensed representation of the alignment of a read to the reference, and is similar to a CIGAR string

* + - 1. CalMD
         1. Calculate MD Tag
         2. Add ‘MD tag’ back to BAM
      2. Sort VCF output
         1. VCFsort

Before continuing, we well sort the variant calls so they are in coordinate order.

This will help us compare against the truth SV record for our simulated isolates (provided SV records are sorted by coordinate),

in future will allow us to view the variants using a genome browser.

* + - 1. AWK
         1. Creating a summary
         2. a lot of the important information is shoved in the ‘INFO’ field.
         3. We would prefer a format which identifies the contig, location, type, and size of each variant call in an easy to read manner.
         4. Result (easy to read): 
      2. Calculating sniffles Performance Metrics
         1. the variant calls provided by sniffles
         2. the ground SV truth
         3. calculate performance metrics for sniffles

The following formulas for accuracy, precision and recall are commonly used when benchmarking bioinformatics software.

Result based on the testing data

While the accuracy and precision of sniffles was good, the recall is low. This is due to a key setting in sniffles which relates to our read set - read support.

* + - * 1. Tuning sniffles Settings

Sniffles has a default setting called ‘read support’ which requires 10 reads to support a possible SV for it to be accepted as genuine

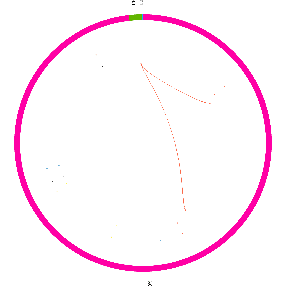
Reducing this number allows more SVs to be discovered, but may also cause some false positives

This value should be based on the coverage of your data

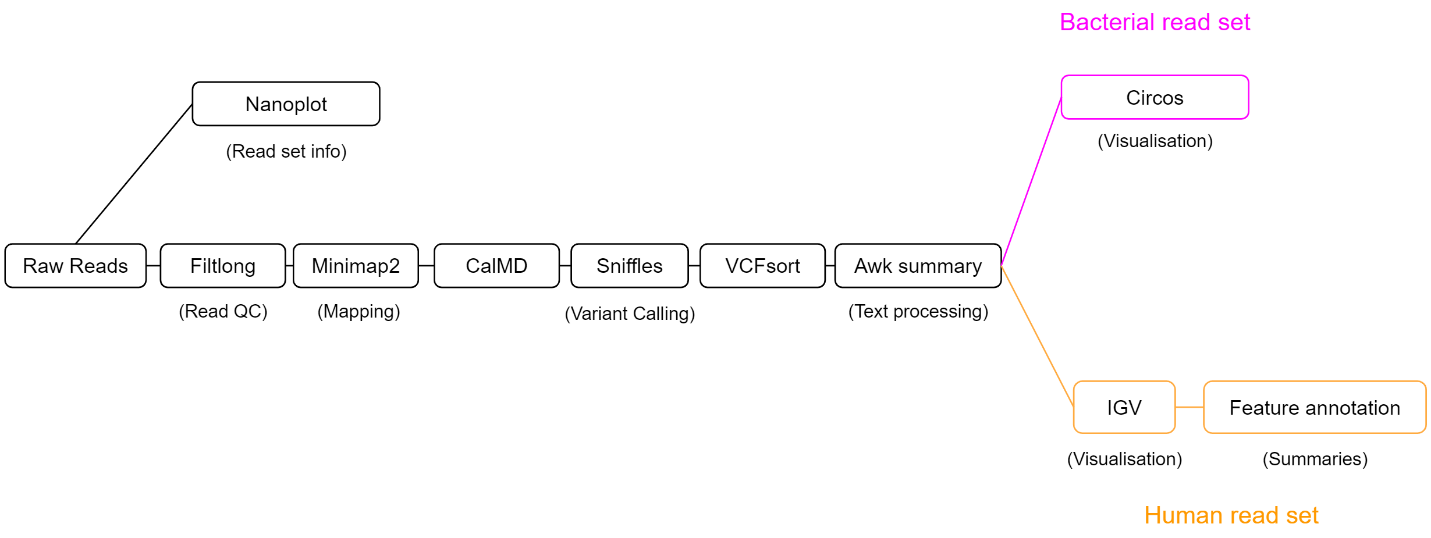
In the testing dataset: As our mean depth is 12x, but the quality of our reads is good, we will reduce the ‘read support’ setting to 5.

* + 1. Feature annotation
       1. Awk
       2. VCFAnnotate
          1. automatically label variants with the genomic features they intersect with
    2. Visualization -> Next
       1. Circos
          1. for the bacterial read set

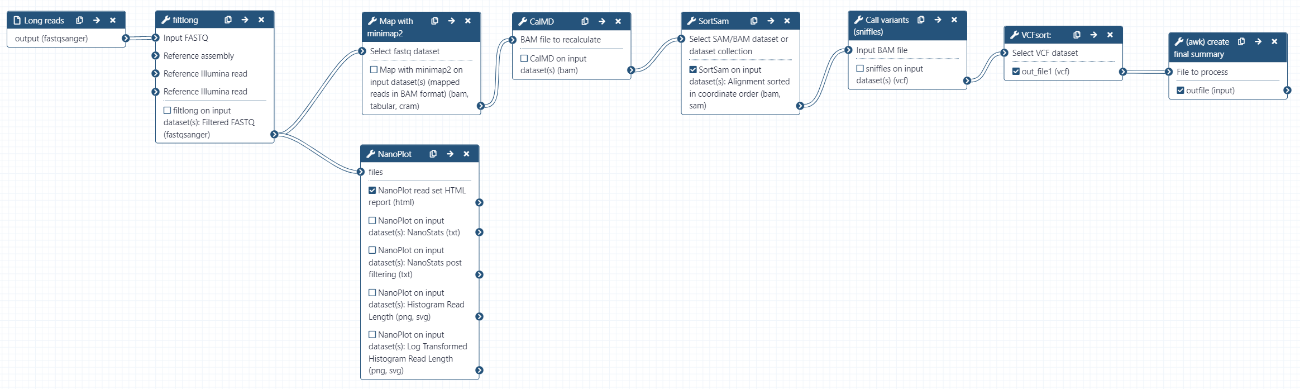
We will be using a workflow to create circos plots for us. This will process our sniffles VCF summary and SV truth report, and produce a plot.



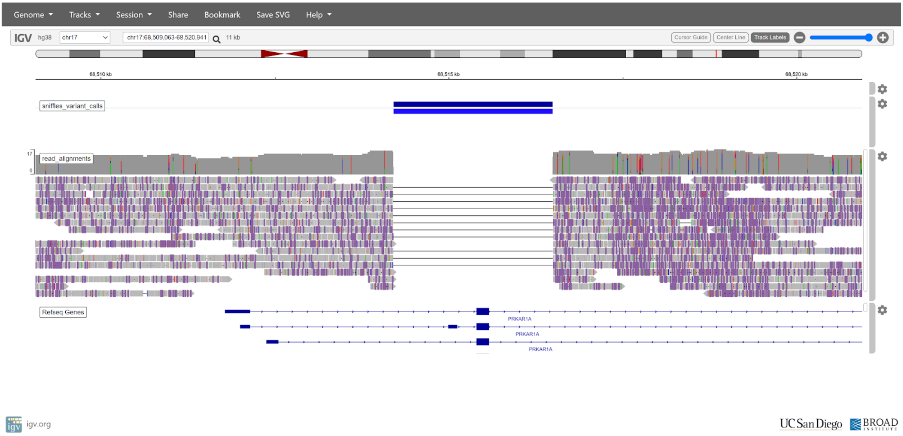
* + - 1. IGV
         1. performed to understand variation on a genome-wide scale



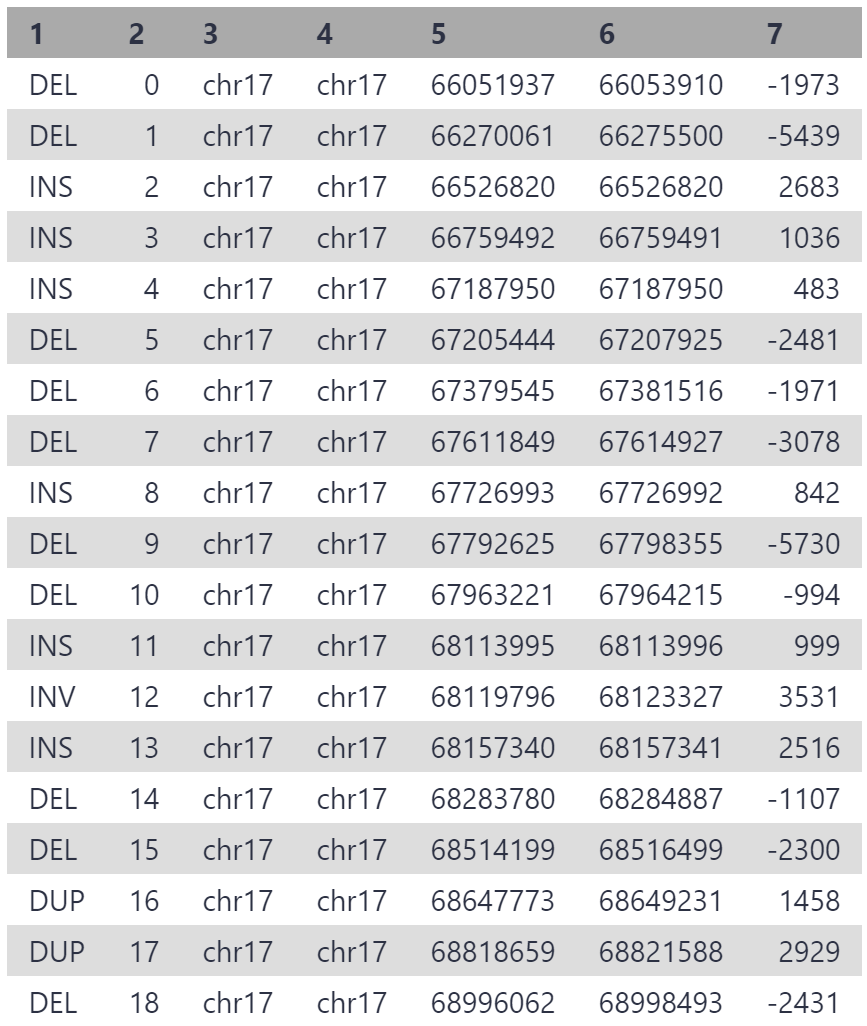
1. Different Dataset
   1. Bacterial Dataset
      1. Three important dataset
         1. A set of reads
            1. isolate\_reads.fastq
         2. A reference genome
            1. ecoli\_sakai.fasta
         3. ‘ground truth’
            1. isolate\_sv\_record.tsv
      2. E. coli sakai assembly
      3. NanoSim
         1. Nanopore reads of this mutated genome were simulated using NanoSim
         2. The benefit to this approach is that we know the ground truth. SVs were manually added to our E. coli sakai reference genome, and their details were recorded
   2. Human Dataset
      1. Sertoli-Leydig cell tumor
         1. long reads. 26.7 Gb of reads were produced using the PacBio Sequel system, equating to an average read depth of 8.6x
         2. we need reads from our section of chr17 (pos 66,000,000 - 69,000,000) for variant calling against hg38
         3. and a file listing genomic features (GFF) for automated annotation later on
      2. workflow
         1. Read QC (Filtlong)
         2. Produce a summary report of filtered reads (NanoPlot)
         3. Align reads to hg38 (minimap2)
         4. Calculate the MD tag (CalMD) and sort the BAM file by coordinate (SortSam)
         5. Call variants (sniffles)
         6. Sort the variant calls (VCFsort)
         7. Create a summary of the variant calls (awk)



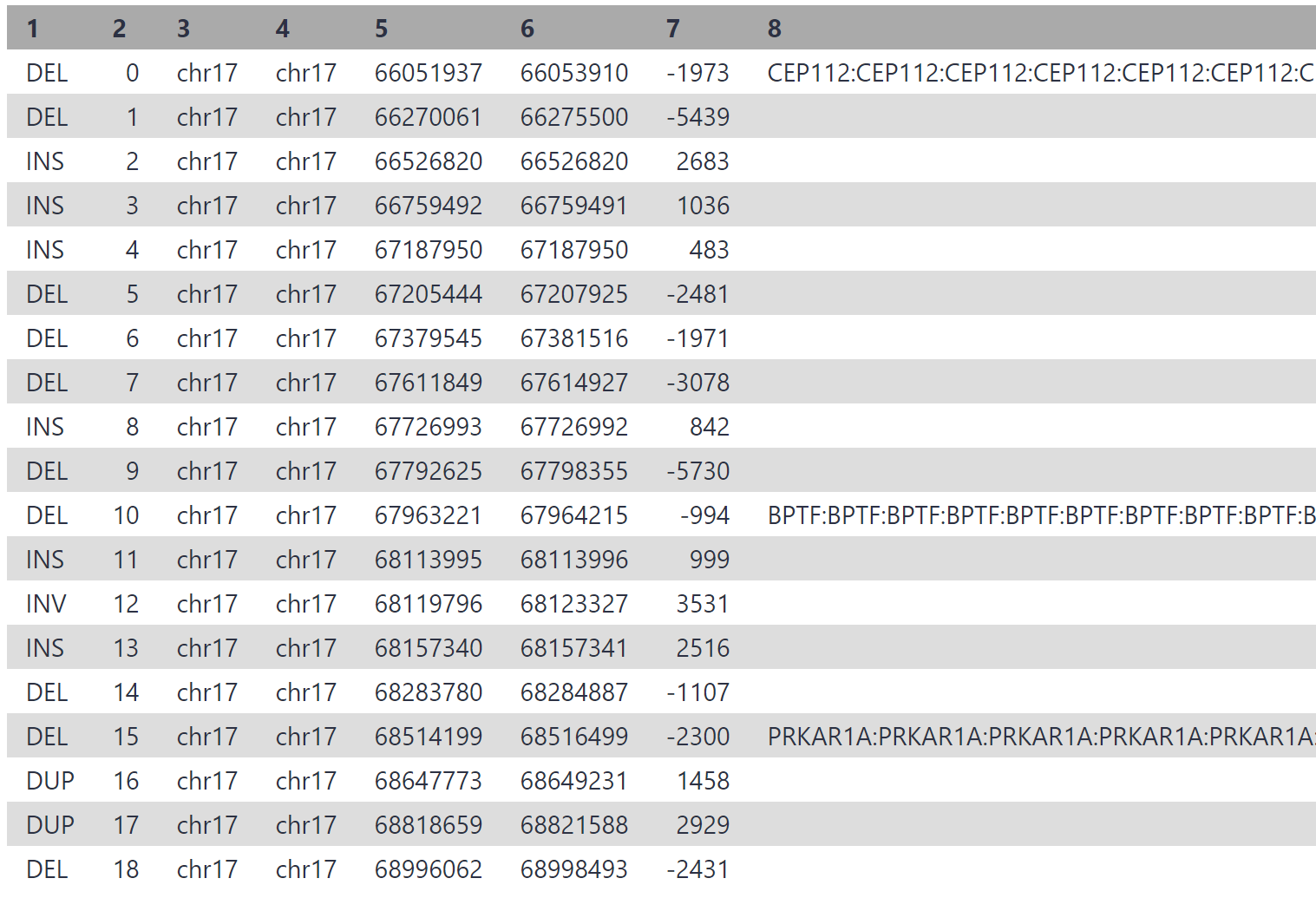
* + 1. The SV calling result in igv



* + 1. Genome annotation
       1. Before



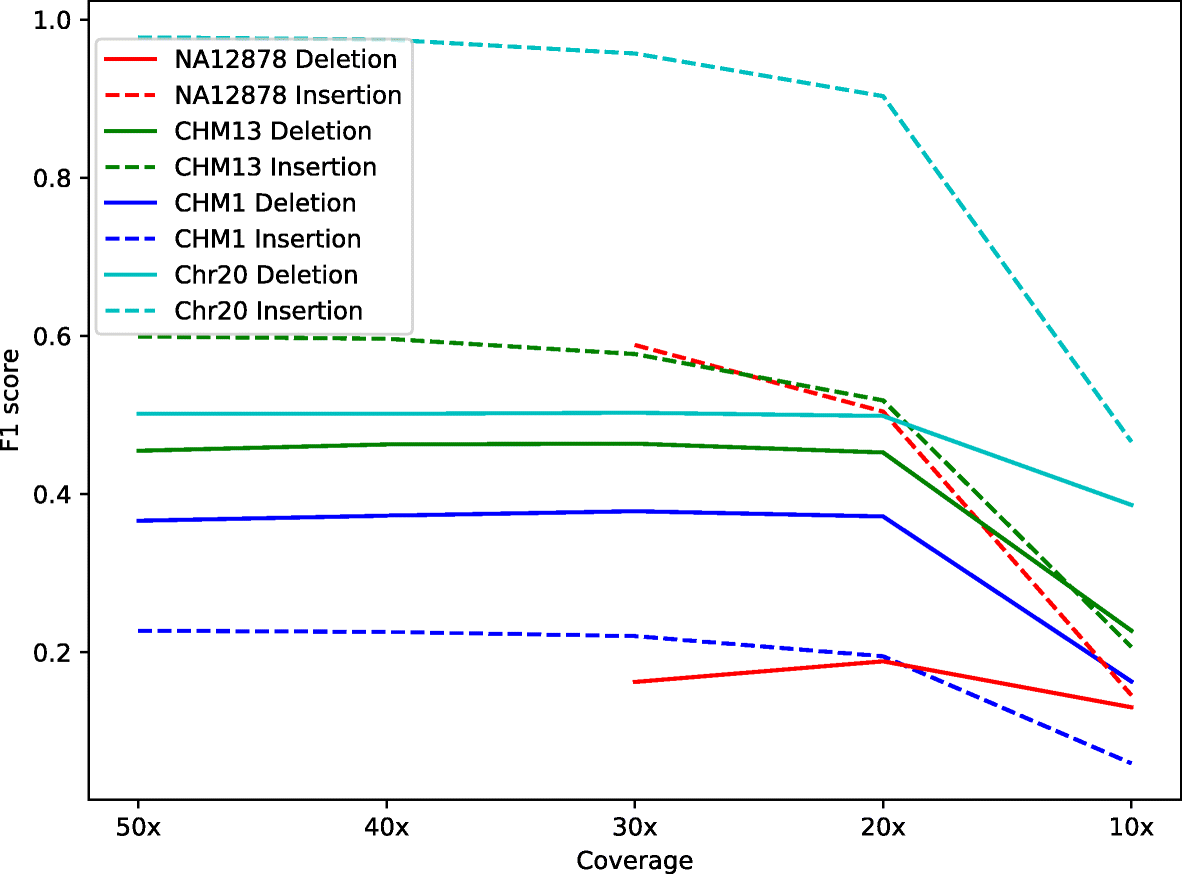
* + - 1. After



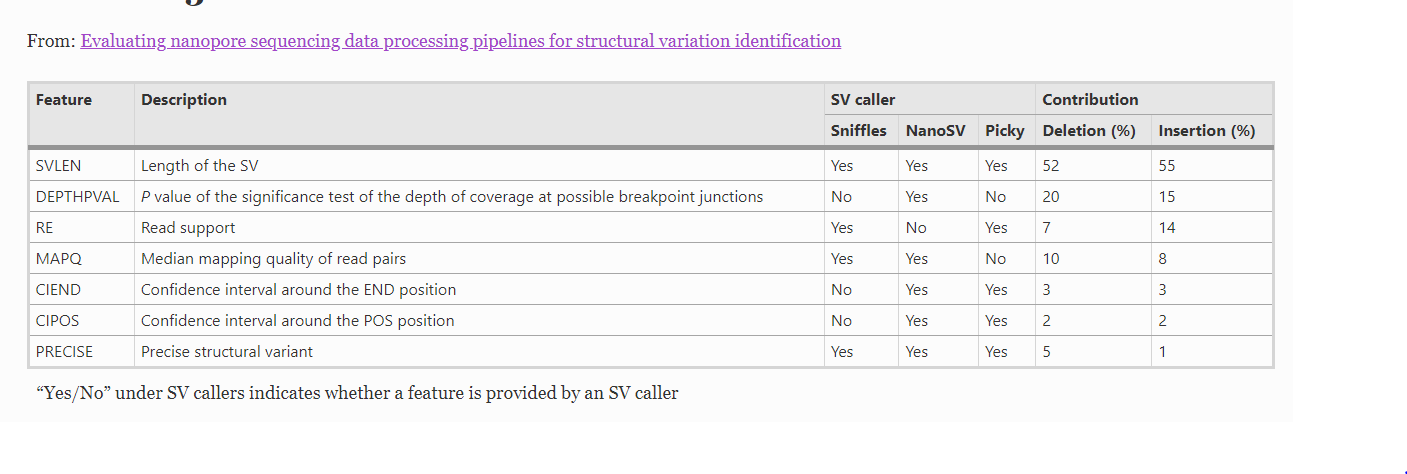
* + - * 1. In the associated paper, authors reduced the initial > 13,500 variant calls down to only 3, by filtering for variants which **overlap a disease gene coding exon**, and **those which are not present in a healthy control sample.**

**2: Evaluating nanopore sequencing data processing pipelines for structural variation identification**

1. <https://link.springer.com/article/10.1186/s13059-019-1858-1> -> check it tomorrow.
   1. Even it is ONT, I think it is still valuable for the pacbio sv detection.
2. Key conclusion
   1. **For an initial data assessment,** we recommend using aligner minimap2 in combination with SV caller Sniffles
      1. because of their speed and relatively balanced performance.
   2. **For detailed analysis,** we recommend incorporating information from multiple call sets to improve the SV call
3. Data
   1. nanopore sequencing of the human samples NA12878 (30x)
   2. CHM13 (referred to as CHM13) (30x)
4. Aligner
   1. Used alingers
      1. GraphMap
      2. LAST
      3. Minimap2
      4. NGMLR
   2. Performance
      1. performed similarly across datasets
      2. minimap2 was the fastest by a large margin compared to other aligners, while GraphMap was the slowest
5. SV caller -> consider usability, compatibility, maintenance status, and popularity.
   1. NanoSV
      1. based on minimap2
      2. consumed substantially more resources than the other two SV callers.
      3. called more insertions and deletions than Sniffles and Picky
   2. Picky
      1. Based on LAST
      2. performed fewer file system operations partially because the “select representative reads” step was already performed in combination with LAST before the SV calling step.
      3. In the simulated Chr20 dataset, Picky called more small deletions than any other pipeline.
         1. This is likely due to the Picky’s goal to maximize sensitivity and the high coverage of the Chr20 dataset resulted in a high false-positive rate.
   3. Sniffles
      1. Based on minimap2
6. Tried multiple combinations between different aligner and SV caller
7. For CHM13
   1. Minimap2-NanoSV had the highest recall rate
   2. GraphMap-Sniffles had the highest precision.
8. For F1 score comparison
   1. NanoSVs and Sniffles each had the highest F1 score in four combinations.
   2. In contrast, LAST-Picky had the lowest F1 scores in six combinations.
9. For coverage
   1. Coverage < 20 will significantly impact the performance of each sv caller
      1. We use 10 as the threshuld for both Sniffle and Picky
   2. Graph

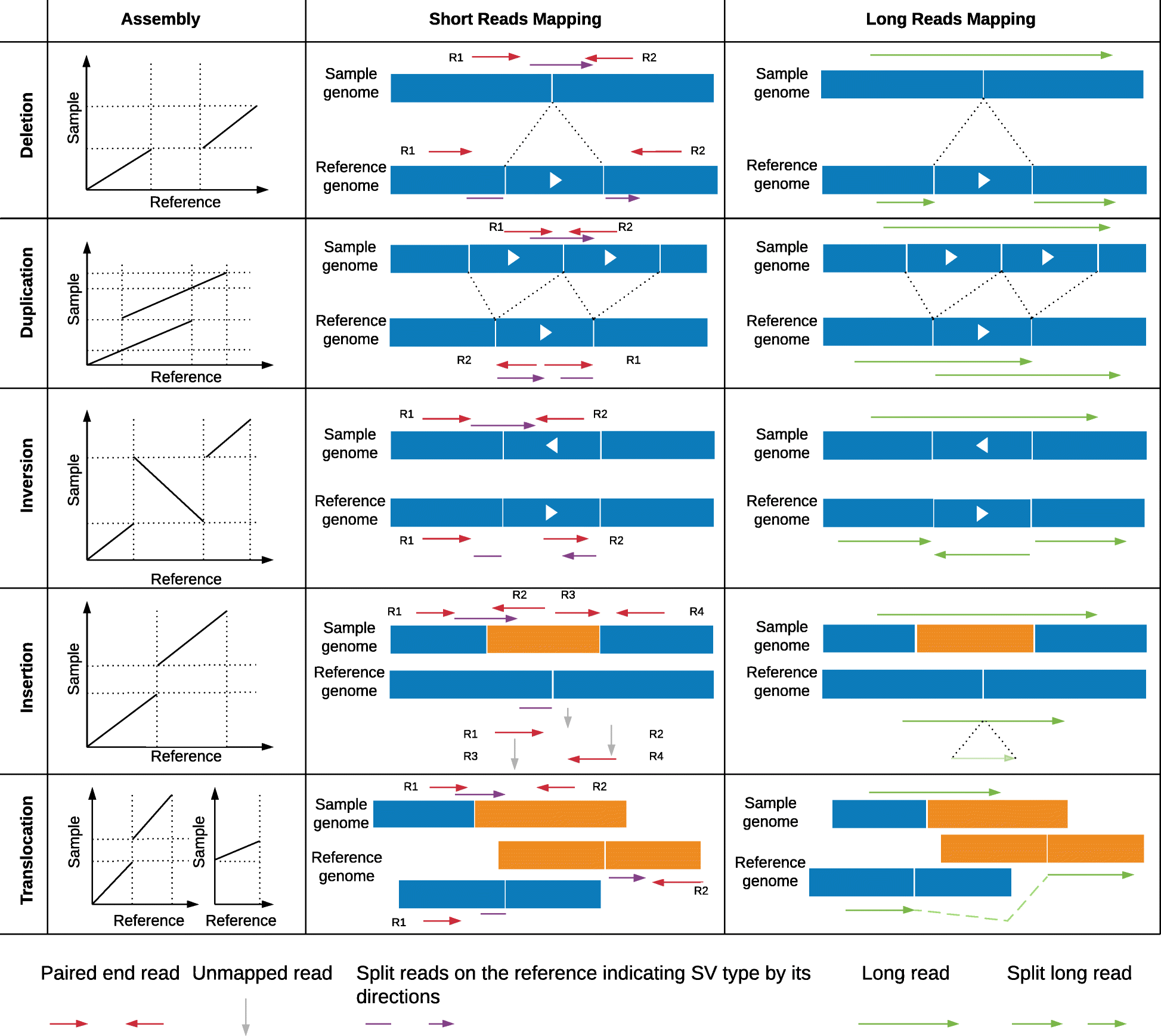


1. How to get the best result
   1. For deletions:
      1. The F1 scores in all four datasets reached a peak when requiring **overlaps of six or seven pipelines**
      2. Notice: each combination is NOT different SV caller but different combination between aligner and SV caller
   2. For insertion
      1. applying the consensus pipeline filter also increased the F1 scores, and calls shared among **two or three pipelines** resulted in the best F1 scores
2. Use Machine learning method to pick out the most trustable results
   1. SV calls from all seven pipelines for each pipeline were combined and labeled “true” or “false” based on whether they overlapped with the corresponding true set. The combined call set was randomly split into a training set (20% of the calls) and a testing set (80% of the calls) using the python package scikit-learn (v0.21.3, parameter “train\_size=0.2”). The labeled SVs were learned and predicted by XGBoost (v0.90) random forest classifier [34] using the features selected from the “INFO” tag in the VCF files (Table 4). Precision and recall rate of the predictions were calculated by scikit-learn metrics.



**3: Structural variant calling: the long and the short of it**

1. Link
   1. <https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1828-7>
2. Why SV is difficult to be detected (compare with SNV)
   1. sequencing and mapping errors blur the patterns.
      1. Indeed, in contrast to SNVs and smaller insertions and deletions, SVs can cover a large portion of a read or even be larger than the read length—which complicates mapping
   2. the patterns induced by the different SV types can be very similar.
      1. For example, it is often hard to distinguish tandem duplications from novel insertions for genomic alignments
   3. multiple SVs can overlap or be nested, giving rise to much more complex mapping patterns than when considered individually
3. Image about how mapping status is different between long reads and short reads



1. Comments for different SV caller by using long reads
   1. With long reads, the SV detection methods are often tailored to the underlying technology—mainly PacBio or Oxford Nanopore.
   2. One exception is Sniffles [5], which employs a parameter estimation in the beginning and thus adjusts itself to the underlying error model.
      1. Sniffles operates on a per read base, also capable of reporting very low-frequency SVs in the sample. This is particularly useful in cancer or in mosaic variation.
      2. Furthermore, Sniffles allows the detection of more complex or adjacent SVs such as inversions flanked by deletions or inverted tandem duplications.
2. For **PacBio**, **three** main specialized methods have been proposed.
   1. PBHoney [60] uses coverage and split read information relying on BLASR alignments.
   2. PacBio structural variant calling and analysis tools (PBSV) is a method developed by PacBio to detect SVs within the range of 20+ bp
      1. Reads supporting a putative SV are used to generate a consensus, which is then re-aligned to the reference genome.
   3. SMRT-SV [61] includes de novo assembly and a specialized genotyping module.
      1. Reads are first aligned to the reference and, subsequently, a local assembly is performed for each multiple kbp window across the entire genome.
      2. The resulting assemblies are then aligned back to the reference, and structural variants (insertion, deletions, and inversions) are identified.
3. For **Oxford** Nanopore,
   1. NanoSV was one of the first methods developed
4. The performance comparison between short reads and long reads
   1. Overall, long-read mapping-based methods for SV calling often show a better performance than short-read ones
   2. Still some performance deficiencies for larger (5+ kbp) insertions compared to de novo assemblies.
   3. Nevertheless, multiple papers have reported a significant improvement in precision and recall for SV calling using long reads compared to short-read mapping approaches

**4: Structural variant detection in cancer genomes: computational challenges and perspectives for precision oncology**

1. Link: [Structural variant detection in cancer genomes: computational challenges and perspectives for precision oncology | npj Precision Oncology (nature.com)](https://www.nature.com/articles/s41698-021-00155-6)
2. Some knowledges
   1. At least 30% of cancers have a known pathogenic SV used in diagnosis or treatment stratification.
   2. ~10+ kb
3. Advantage
   1. long reads allow for haplotype phasing of variants and de novo assembly of complex rearrangements
4. Disadvantage (insufficient)
   1. However, haplotyping information is lost as a result of using the consensus of reads with mixed molecular origin. This makes the consensus sequence unsuitable for variant phasing or for studying intra-tumor heterogeneity or polyploidy.
5. Some suggestions
   1. Preliminary comparisons suggest that NGMLR and minimap2 perform well and both algorithms are designed to handle the higher error rates and adjust for the 1 bp indels in long-reads
6. Different types of method
   1. Alignment-based SV detection algorithms for long-read data
7. Evaluate different tools
   1. At present only nanomonsv reports somatic SVs from long-read data
   2. The commonly used tools SVIM and Sniffles have shown good precision and sensitivity in multiple performance assessments
      1. They were among the first to process both ONT and PacBio data despite their different error profiles and have been followed by additional tools like NanoVar and CuteSV
8. how to collect the final result from different call set
   1. (1) Similar to short-read tools, using a consensus callset created by intersecting multiple long-read SV detection algorithms can increase precision32,67.
   2. (2) Alternatively**, machine learning** approaches can attain greater improvements in precision and sensitivity than ad hoc intersection, given a truth set is available for training3 (these method is better than vote or whatever)
9. mentioned other method: hybrid, or RNA + WGS
   1. however, these are not related to our senarios.

**5: SVIM: structural variant identification using mapped long reads**

(1) VIM consists of three components for the collection, clustering and combination of structural variant signatures from read alignments

(2) It discriminates five different variant classes including similar types, such as tandem and interspersed duplications and novel element insertions.

1) Deletions

2) Interspersed Duplications

3) Novel Insertions

4) Inversions

5) Tandem duplications

(3) works for both Pacific Biosciences and Nanopore sequencing machines.

(4) Advantages

a) Better to find the mobile element

i) Details: All three methods regard SV (i.e. deletions, insertions, inversions) as rearrangements occurring in a single genomic locus. However, SV often involves multiple genomic loci, such as for a mobile element which is reverse-transcribed from a source region and inserted at another location. The higher read lengths of PacBio and ONT reads allow to link both loci much more efficiently and confidently than was possible with short paired-end reads. Nevertheless, existing methods ignore this type of information and are only able to detect the isolated destination location of the mobile element insertion.

b) higher recall and precision than existing tools for SV detection from long reads.

c) SVIM has been specifically designed to distinguish three separate classes of large insertions

1) interspersed duplications,

2) tandem duplications

3) insertions of novel elements

\* Notice: it is the only tool capable of identifying not only the insertion location of an interspersed duplication but also its potential genomic origin using long reads

(5) Compared with other three methods, including

a) *PBHoney-Spots*

*b) PBHoney-Tails*

*c)*  *Sniffles*

**6: Jasmine: Population-scale structural variant comparison and analysis -> Go next week**

**7: Conclusion**

1. Key steps
2. Tools used in each step
3. Pipeline implementation

**8: Tools summarization**

1. Tools: NanoPlot, Filtlong, minimap2, CalMD, SortSam, sniffles, VCFsort, VCFannotate, awk, Circos, IGV

**9: reference**

1. Structural variant calling - long read data (workshop, The university of Melbourne)

<https://www.melbournebioinformatics.org.au/tutorials/tutorials/longread_sv_calling/longread_sv_calling/>

**10: Key knowledges**

1. SV
   1. > 50 bp
   2. Insertion, Deletion, reversion, duplication (interspersed, tandem), translocation (inter, intra-chromosomal)
      1. Many SV calling programs do not attempt to detect **interspersed duplications** and **intra-chromosomal translocations**. This is because most SV callers have been designed for use with human data, and these variants are less relevant to the human genome than other organisms.
         1. interspersed duplications is also not that common (so less important)
         2. Intra-chromosomal translocations are ignored for similar reasons.

1. Some valuable statistical results
   1. This is highly pertinent when working with the human genome, as more than 60% appears to be repetitive sequence.
      1. mobile elements
      2. repeats.
2. Destructive (more interesting) and non-Destructive