

Structural Variation and indel Discovery

Fereydoun Hormozdiari

Eichler Lab

Genome Sciences, University of Washington

Outline

- Structural Variation and CNV
- Read Depth Signature
- Read Pair Signature
- Split Read Signature
- Multi Signature
- Assembly based methods
- Toy Exercise

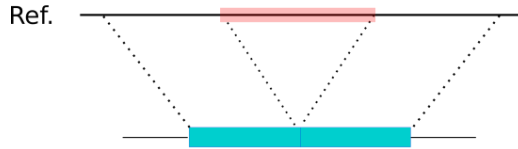
Structural Variations

“Genomic rearrangements that affect >50 bp of sequence, including deletions, novel insertions, inversions, mobile-element transpositions, duplications and translocations.”

Alkan et al. 2011. NRG

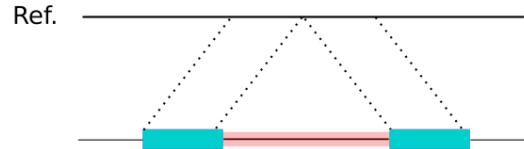
Structural Variation Classes

DELETION

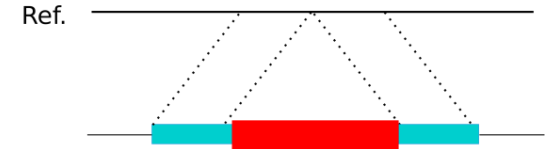


Autism, mental retardation, Crohn's

NOVEL SEQUENCE INSERTION



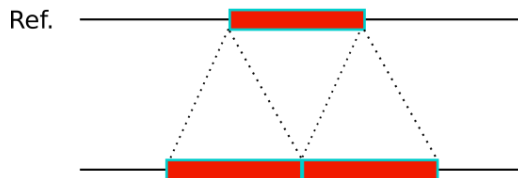
MOBILE ELEMENT INSERTION



Alu/L1/SVA

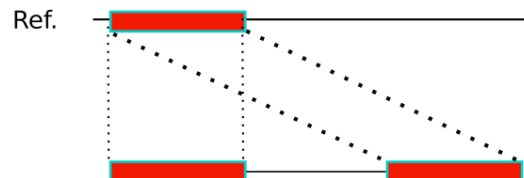
Haemophilia

TANDEM DUPLICATION



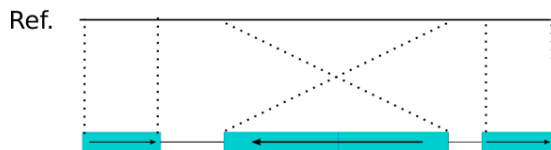
Schizophrenia, psoriasis

INTERSPERSED DUPLICATION



CNV: Copy number variants

INVERSION



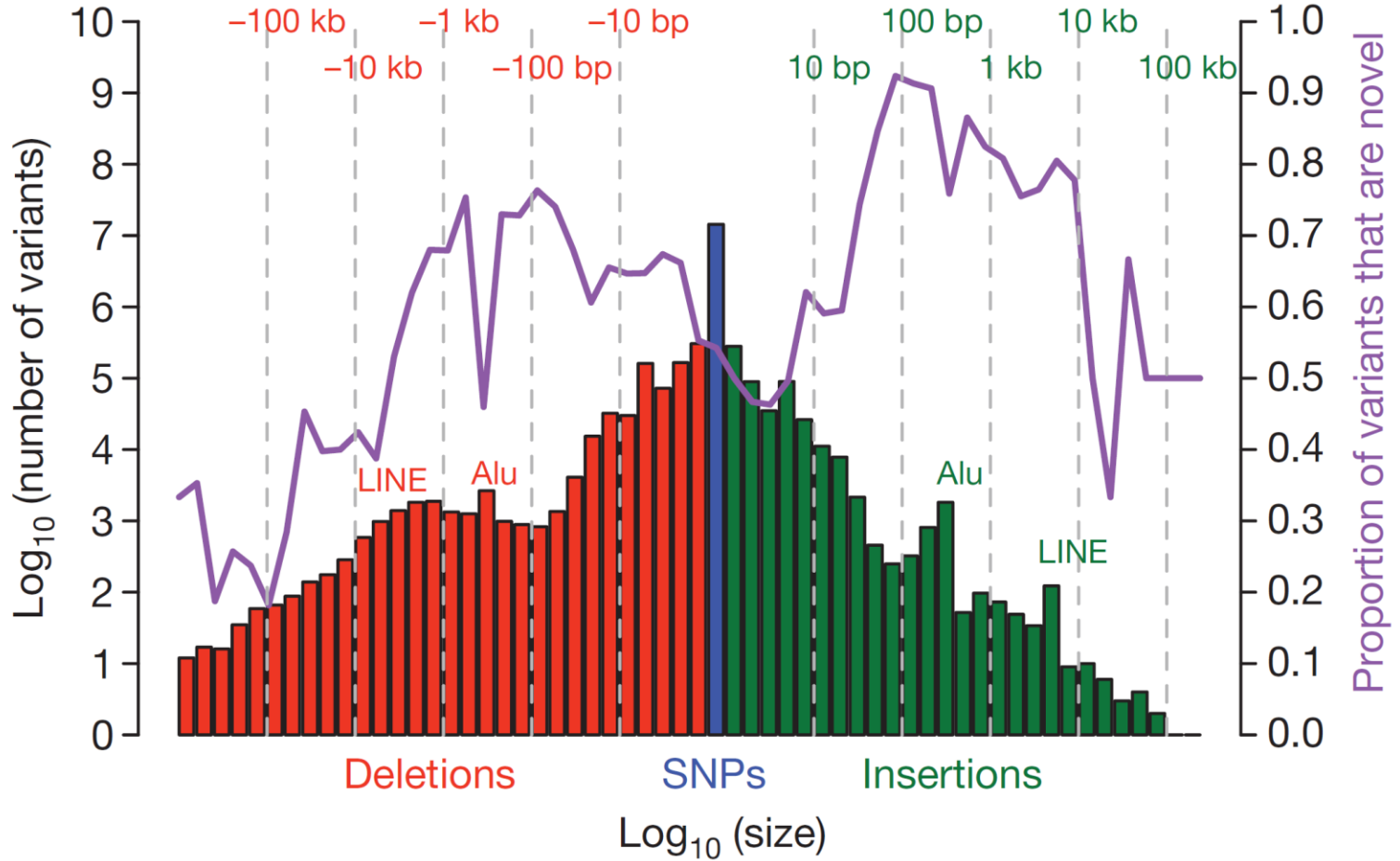
TRANSLOCATION



Chronic myelogenous leukemia

Balanced rearrangements

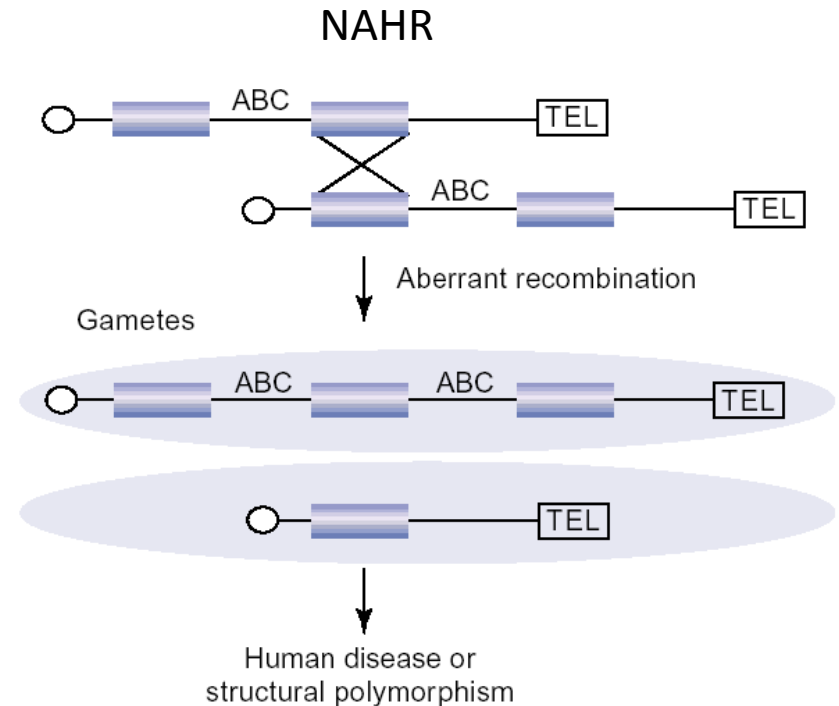
Abundance of CNV



Mechanism of Deletion and Duplications

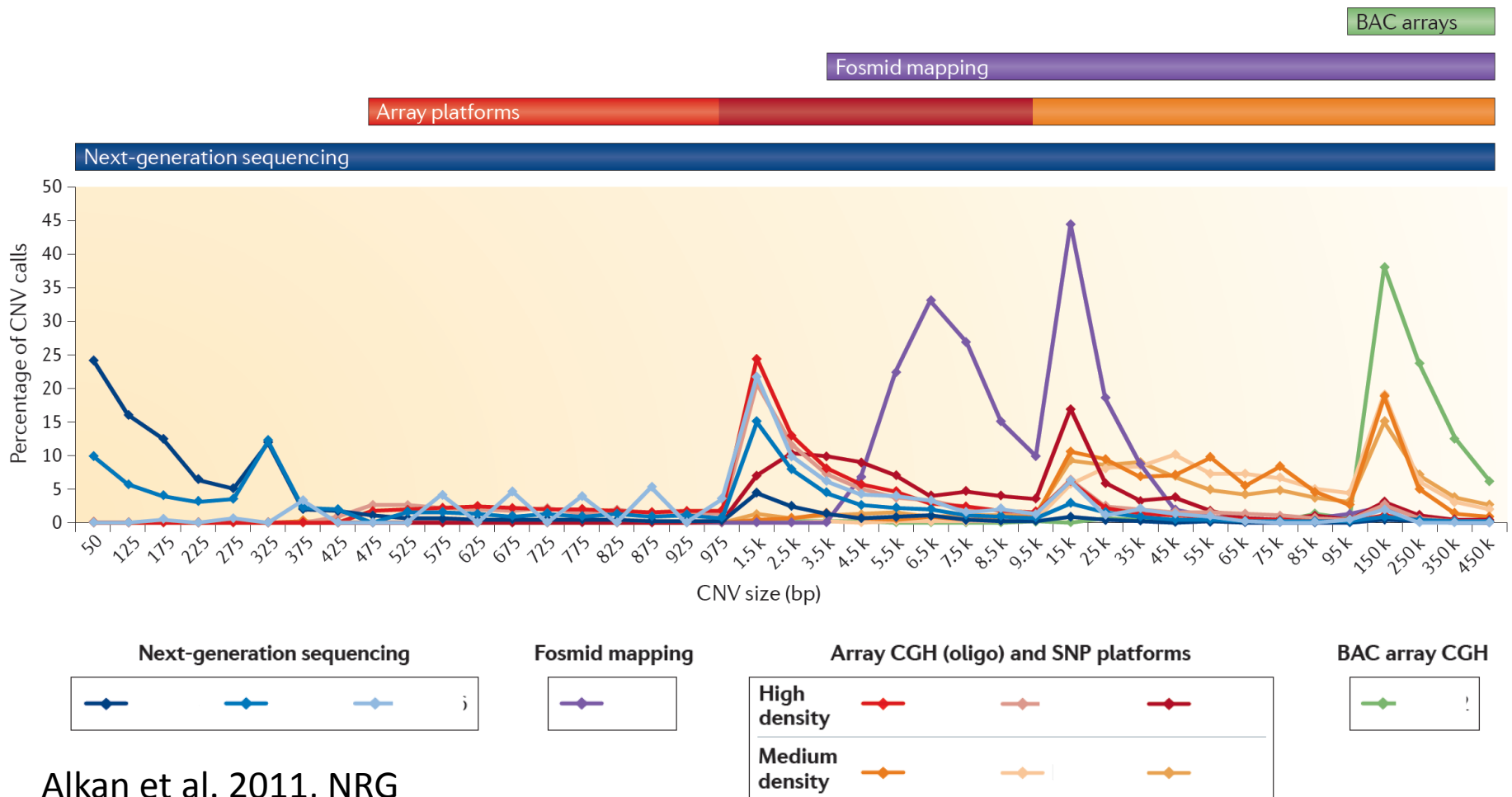
Mechanism

- Non-allelic homologous recombination (NAHR)
- Non-homologous end joining (NHEJ)
- Variable number of tandem repeats (VNTRs)
- Mobile element insertions (MEI)
- ...



Power of next generation sequencing

Variable power of different platforms. NGS although more expensive but gives the Best resolution for CNV discovery.



Structural variation discovery with NGS data

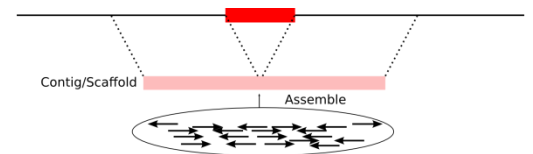
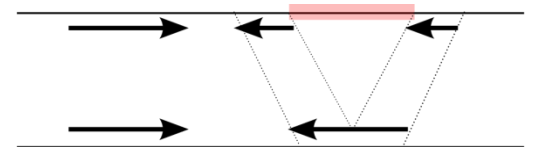
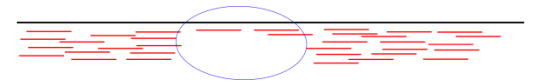
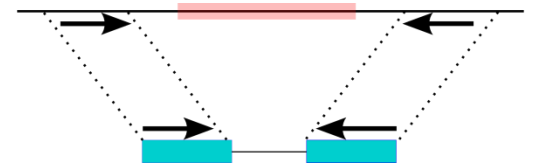
- ❑ SVs: genomic alterations > 50 bp.
 - Databases:
 - dbVar: <http://www.ncbi.nlm.nih.gov/dbvar/>
 - DGV: <http://projects.tcag.ca/variation/>
- ❑ Input: sequence data and reference genome
- ❑ Output: set of SVs and their genotypes (homozygous/heterozygous)
- ❑ Often there are errors, filtering required
- ❑ SV detection methods can be based on statistical analysis or combinatorial optimization
- ❑ Tools: VariationHunter, BreakDancer, MoDIL, CommonLAW, Genome STRiP, Spanner, HYDRA, etc.

Challenges

- Most SVs are embedded within or around segmental duplications or long repeats
 - If you use unique mapping, you will lose sensitivity
 - Ambiguous mapping of reads will increase false positives
 - Reference genome is incomplete; missing portions are duplications which cause more problems in accurate detection
- Many SVs are complex; many rearrangements at the same site
- CNV discovery is heavily studied but still not perfect; detection of balanced rearrangements are still problematic


Sequence signatures of structural variation

- Read pair analysis
 - Deletions, small novel insertions, inversions, transposons
 - Size and breakpoint resolution dependent to insert size
- Read depth analysis
 - Deletions and duplications only
 - Relatively poor breakpoint resolution
- Split read analysis
 - Small novel insertions/deletions, and mobile element insertions
 - 1bp breakpoint resolution
- Local and *de novo* assembly
 - SV in unique segments
 - 1bp breakpoint resolution



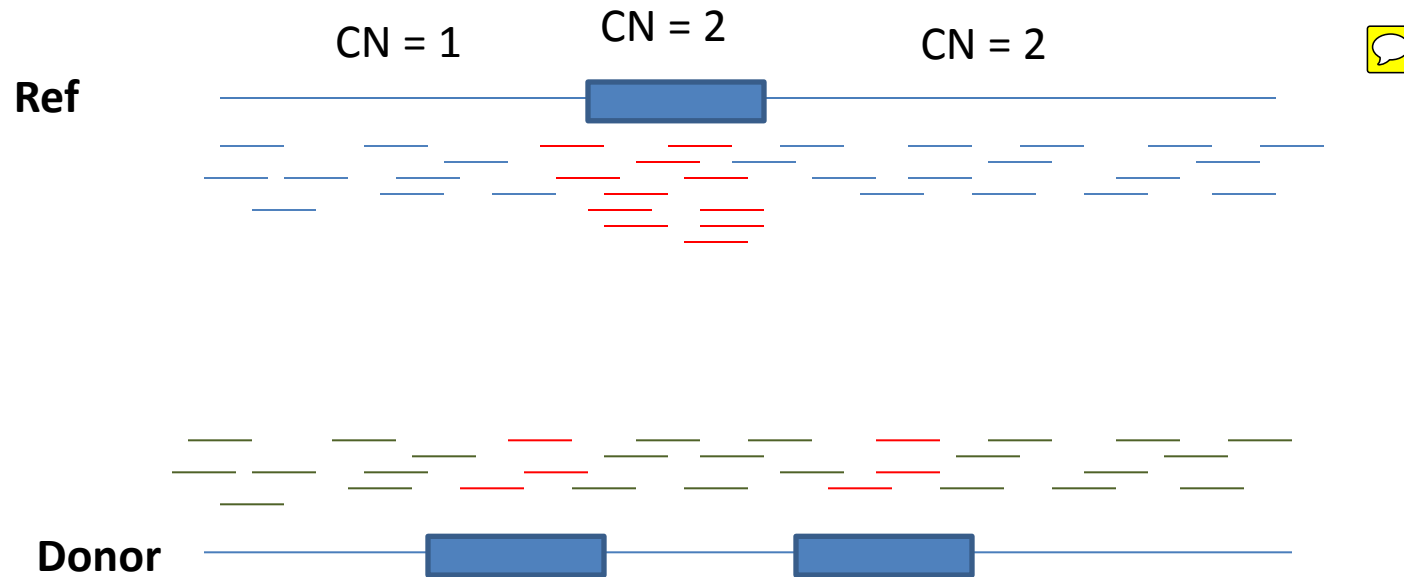
READ DEPTH APPROACH

Read depth based algorithms

- Assume Poisson distribution of reads sequenced across genome
- Multiple mapping:
 - WSSD (whole genome shotgun sequence detection) [*Bailey et al., Science 2002; Alkan et al. 2009 Nature Gen.*] 
- Unique mapping:
 - Low resolution: Campbell et al. Nat Genet 2008, Chiang et al. Nat Meth, 2009 (SegSeq)
 - High(er) resolution: CNVnator, EWT (RDXplorer)

Read Depth Approach

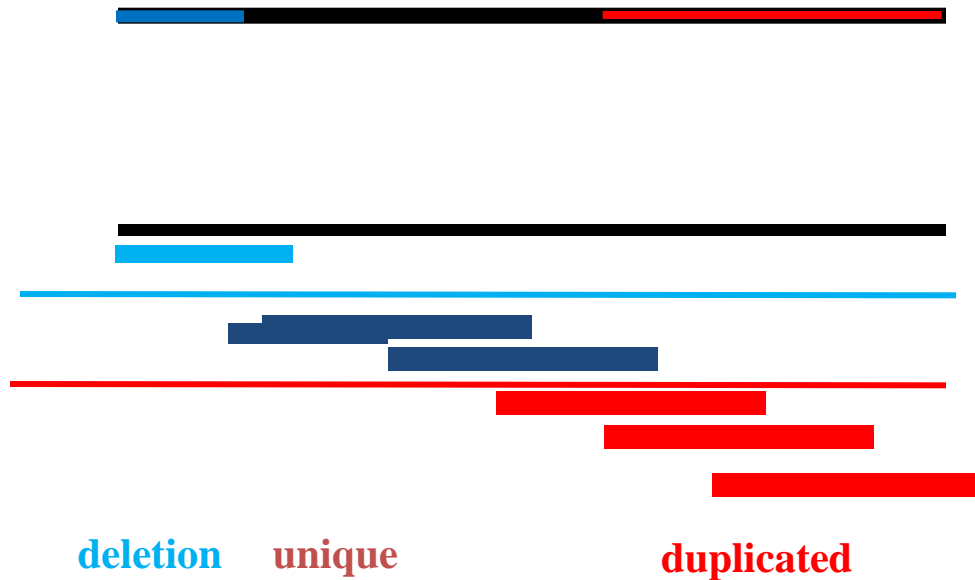
- Using the read depth signature we can predict the copy number of each region in a given donor genome.



Read depth analysis: WSSD

- Uses database of random reads to confirm duplicated nature of the sequence
 - increased # of copies => increased number of reads
 - decreased # of copies => decreased number of reads
- Compute depth-of-coverage in 5kb windows (sliding by 1kb); select regions with increased depth as **duplications**, regions with reduced depth as **deletions** (WSSD method)

Sequence to Test



Random Genome Sample (Whole-Genome Shotgun Sequence)



Multiple vs. unique mapping

Genome

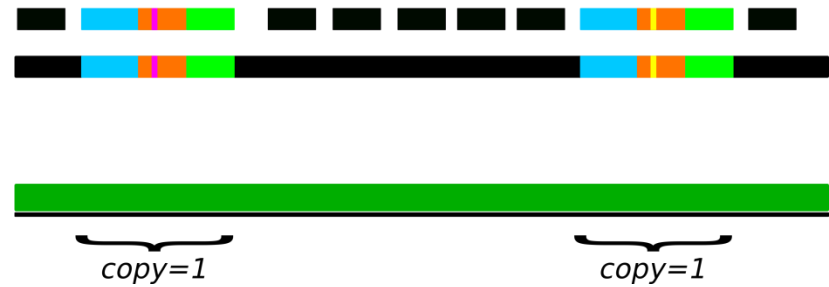
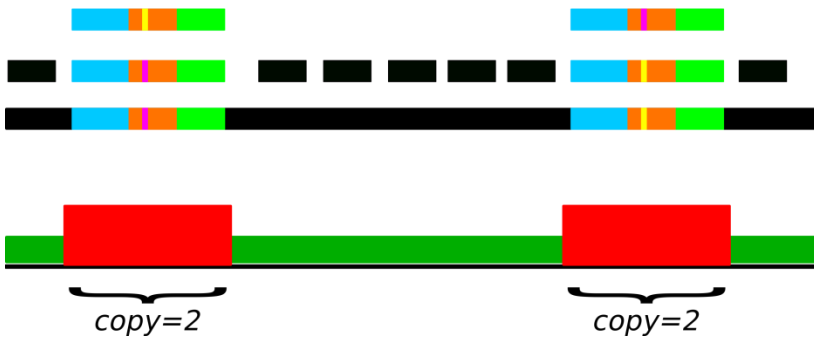
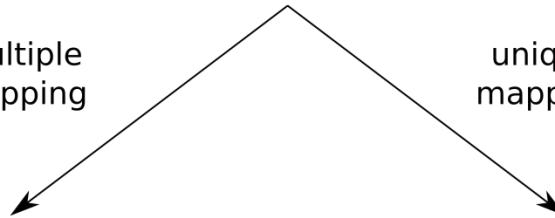


Reads



multiple
mapping

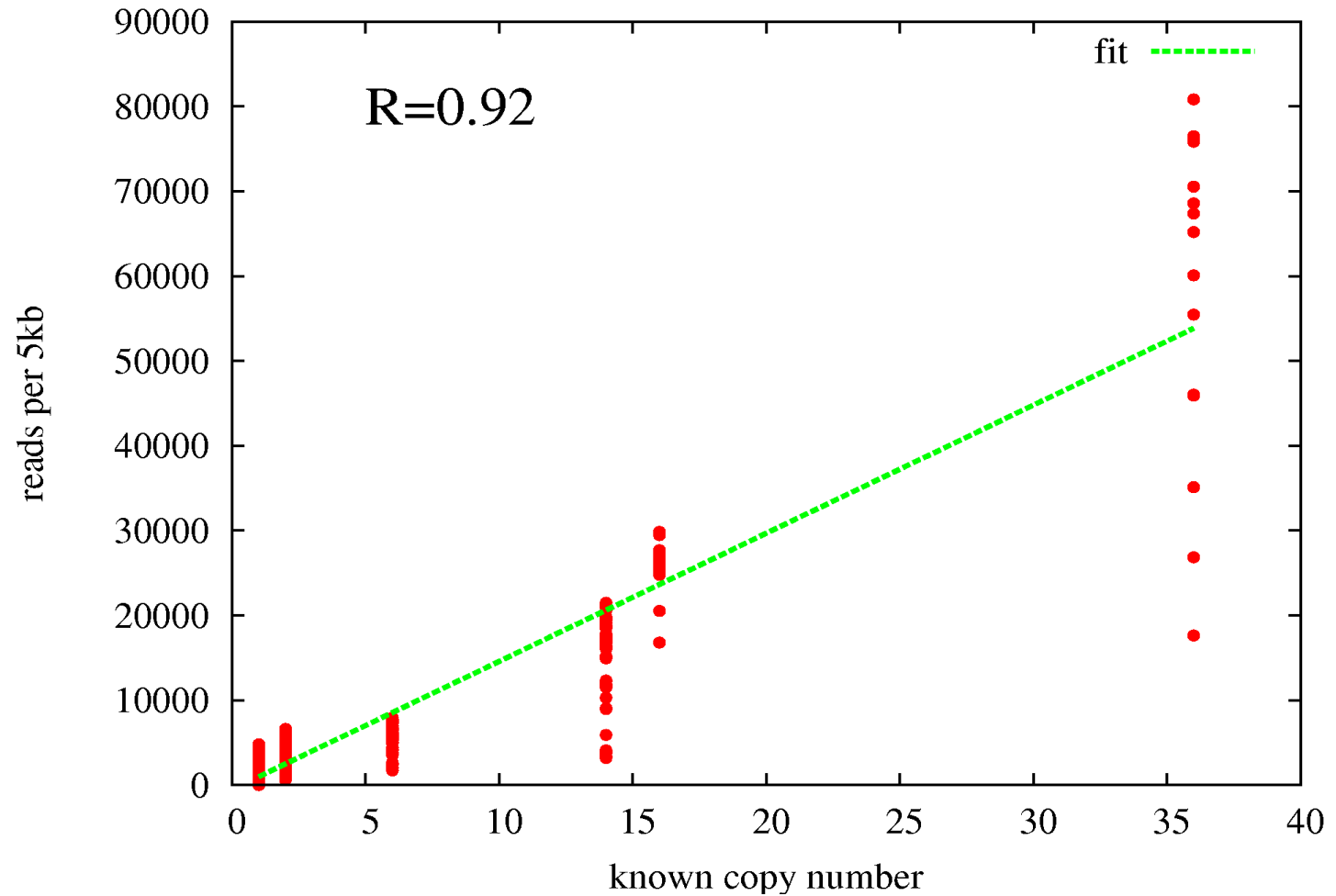
unique
mapping



WSSD: next-gen

- NGS specific problems
 - Short reads: MegaBLAST is replaced by mrFAST / mrsFAST
 - Common repeats: all repeats need to be masked
 - GC % bias needs to be fixed
- Improvement
 - Absolute copy number detection in 1 kb non-overlapping windows
 - Genotyping highly identical paralogs

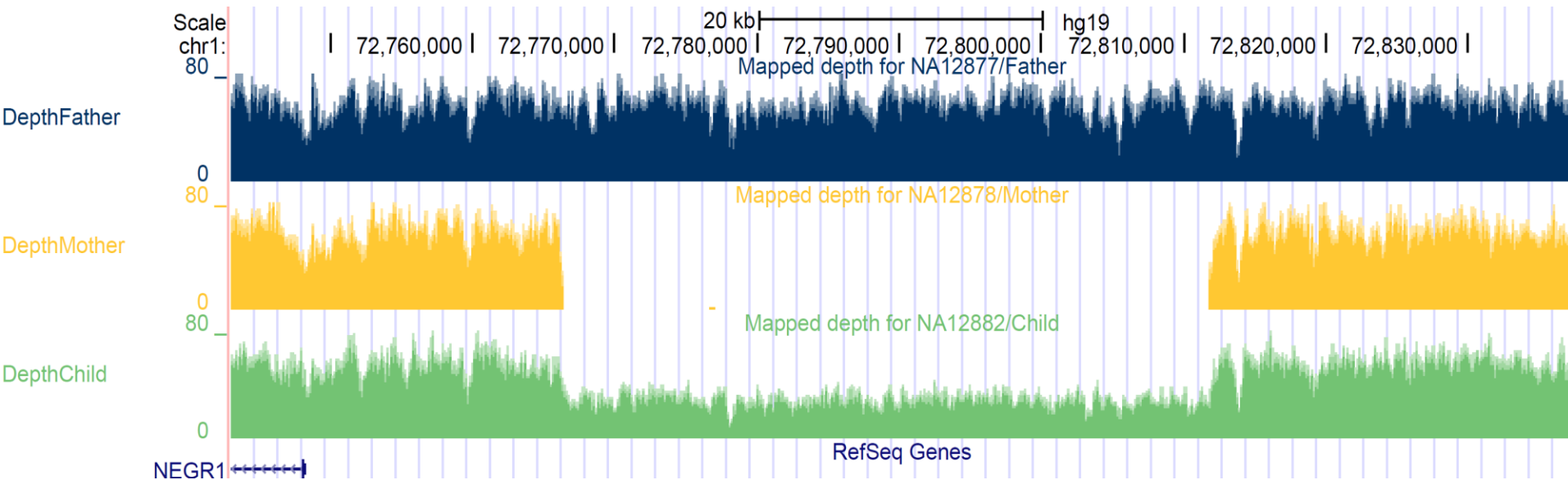
Read depth - Copy number correlation



Real Example – NA12882 trio

Deletions in NA12882 trio (NA12877-NA12878- NA12882) using the high coverage platinum genomes (<http://www.illumina.com/platinumgenomes/>).

Loci: chr1:72,760,000-72,830,000



Read Depth Approach

Pros

- Very accurate and powerful in *large and medium* CNV (>5Kbp) discovery in *unique* regions of genome
- Relatively fast

Cons

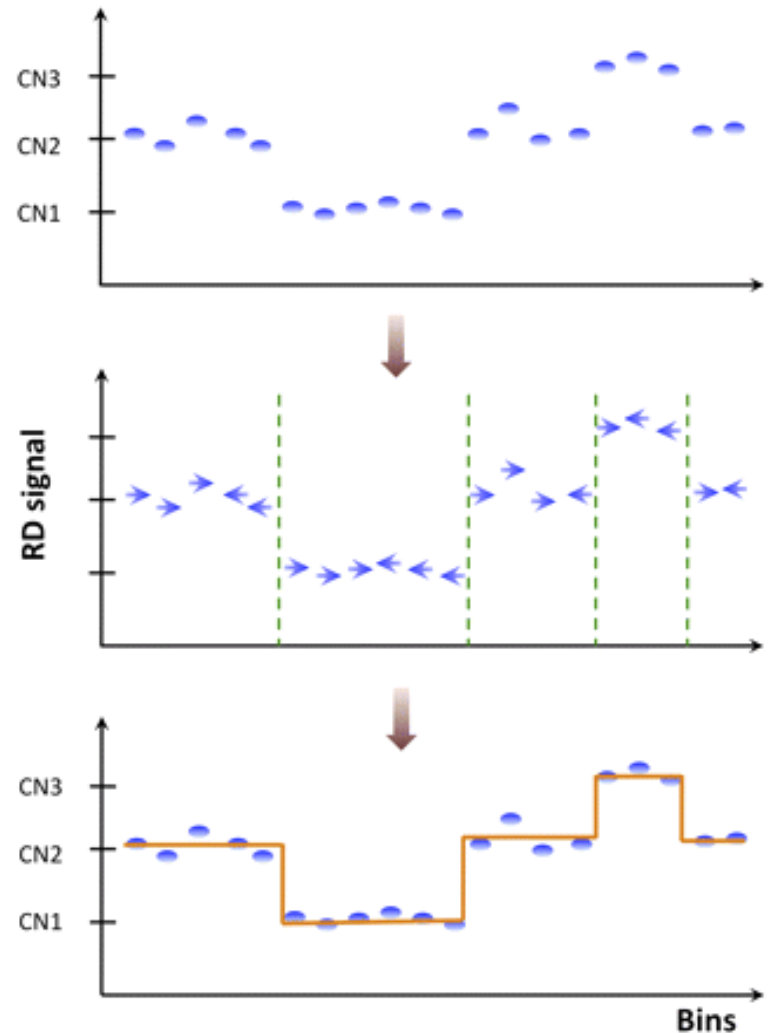
- The false discovery rate (FDR) increases as length of CNV decreases
- The FDR increases in regions of higher CN (like segmental duplications)
- Sequencing biases, such as GC bias, can have negative effect.

Read Depth Based Approach Methods

- **mrCaNaVar** (WSSD for NGS): Uses multiple mappings produced (by mappers such as mrsFast) to detect CNV + GC correction.
- **CNVnator**: Does multiple sample CNV discovery. Might be method of choice for trio studies.
- **CNVer**: predicts the CNV of a region and using a graph flow approach tries to filter false calls. Also in some cases can give the exact breakpoint.
- **Cn.Mops**: used mixture of poisson model and has maximum a posteriori objective function. Can benefit from multiple samples.
- **CopySeq**: Uses additional read-pair information to find more accurate breakpoints.
- ... (Many more)

CNVnator

- Unique mappings
- Mappings with low MAPQ are discarded
- Partitioning is based on mean-shift technique developed for image processing



CNVnator Input and Steps

- Uses BAM or SAM as input.
- Steps:
 - Extracting read mappings from bam/sam file (to separate each region/chromosome for less memory usage)
 - Generating histograms and calculating statistics
 - Read Depth signal partitioning
 - CNV calling

mrCaNaVaR (WSSD) Input and Steps

- Mask all the repeat regions of the genome.
- mrsFast mappings (single end) with edit distance equal to 5% of the read length.
- Read Step: Calculate Read Depth using the above mappings.
- Call Step: Calls the CNV using the segments created in previous step.

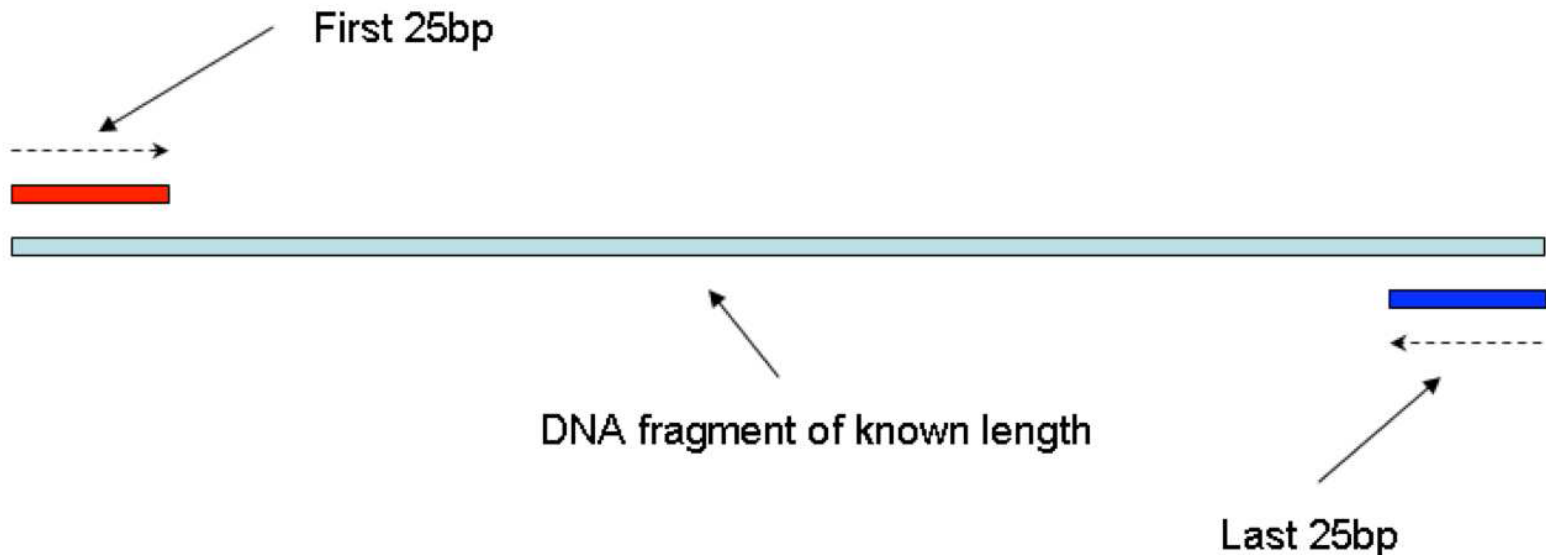
CNVs with exome sequencing

- Exome sequencing: capture only coding exons from DNA and sequence
 - 1% of total genome
 - Good for protein coding variants but misses regulatory sequence, introns, etc.
- Whole genome sequencing generates random data, but exome does not
- Capture efficiency changes for *every* exon (n~200,000)
- CNVs from exons: ExomeCNV,XHMM or CoNIFER.

READ PAIR SIGNATURE

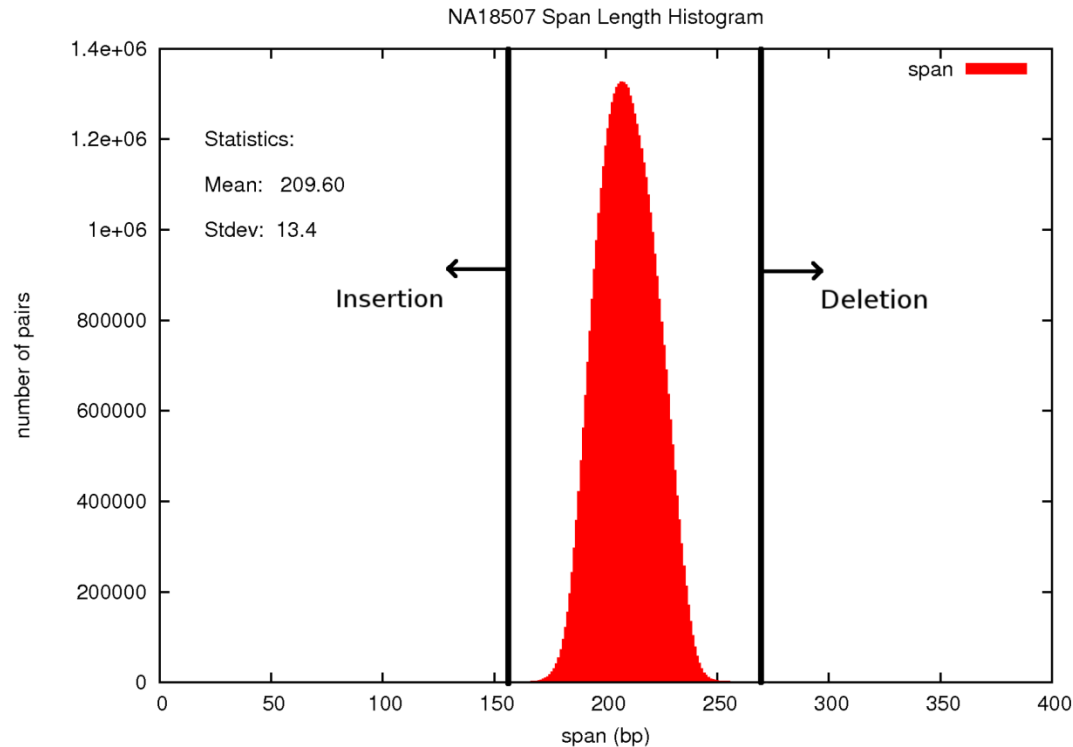
Read Pair Approach Assumptions

- Each fragment of genome is sequenced from both ends (the fragment size falls in a tight normal distribution).
- In most methods it is assumed that the size of fragment has a max/min length $[\Delta_{\min}, \Delta_{\max}]$



Fragment length distribution

- Example of fragment length distribution in NA18507 (sequenced by Illumina)



Concordant = read pairs that map in expected orientation & size

Discordant = read pairs that map different than what is expected

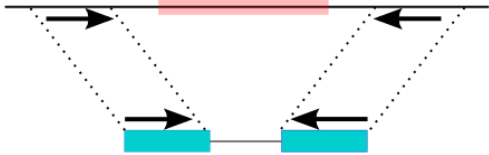
$$[\Delta_{\min}, \Delta_{\max}] = [155bp, 264bp]$$

How to calculate the min/max fragment length

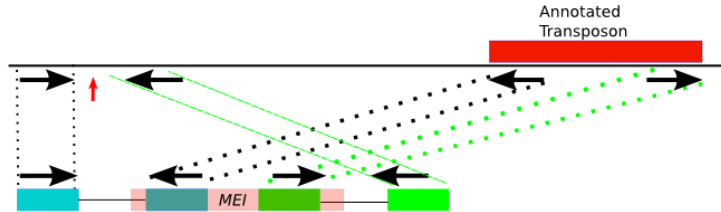
- The most accurate way to calculate the min/max fragment size is using the unique mapped reads.
- Consider the distance between two ends of each paired-end read mapped in correct orientation (FR) in BAM file.
- Filter the largest 1% values (as they are discordant mappings).
- Calculate the mean (μ) and std (σ) of the remaining values
- Finally $[\Delta_{\min}, \Delta_{\max}] = [\mu - 3\sigma, \mu + 3\sigma]$

Read Pair analysis

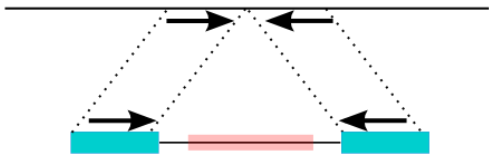
Deletion



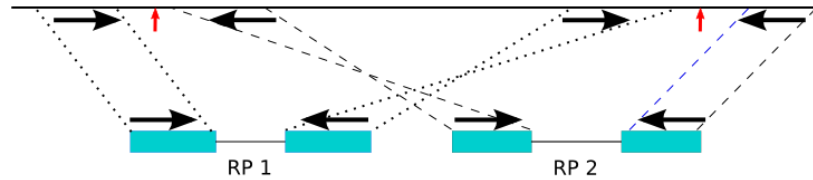
Mobile Element Insertion



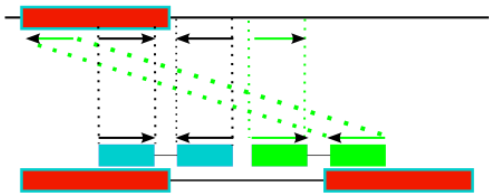
Novel Sequence Insertion



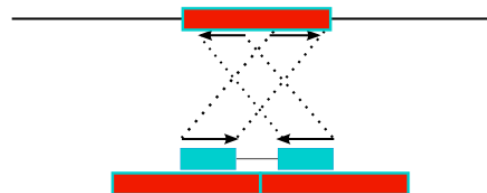
Inversion



Interspersed Duplication

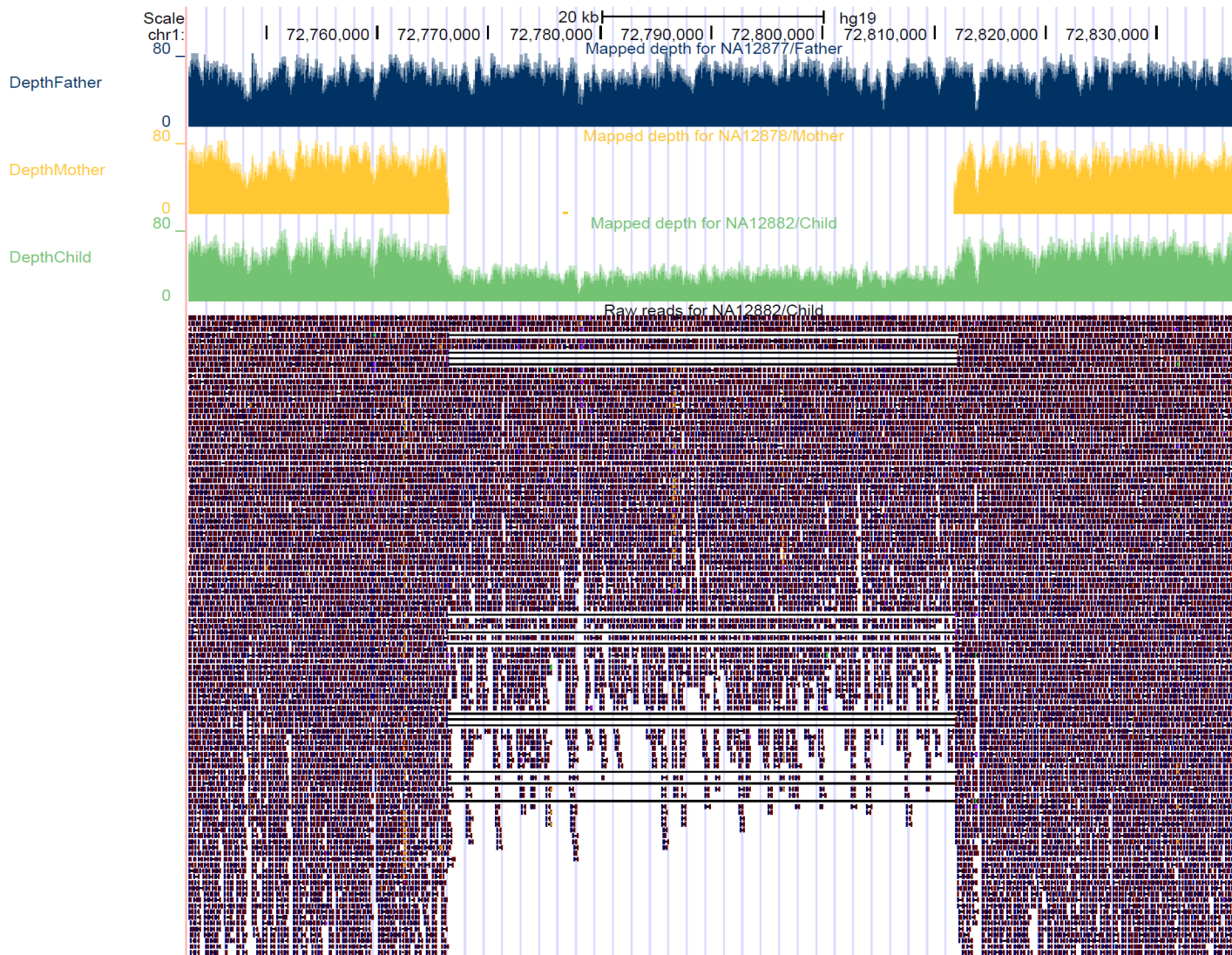


Tandem Duplication



Read Pair Approach Example

The same deletion in NA12882 showing both Read Depth and Read Pair signature (chr1:72760000-72820000)



Platinum Genomes
sequenced by
Illumina


Read Pair Algorithm



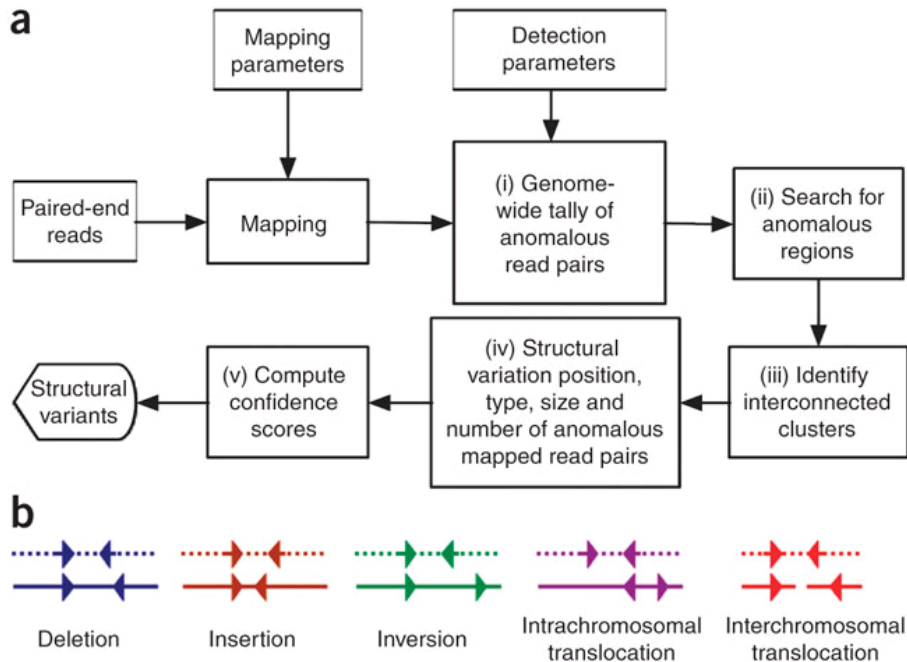
General Read Pair Algorithms strategy


- 1) classify each reads mapping as concordant or discordant.
Concordant reads are reads which both ends are mapped in correct orientation and the distance between two ends is between $[\Delta_{\min}, \Delta_{\max}]$
- 2) Clusters the discordant reads that can support the same SV (deletion, insertion, inversion,...)
A set of discordant reads which a pair of SV breakpoints matches their property.
- 3) Call the potential SVs with high support and low edit distance

Read pair based SV callers


- Unique mapping:
 - BreakDancer, GenomeSTRiP, SPANNER, PEMer (454), Corona (SOLiD), etc.
- Multiple mapping:
 - VariationHunter, CommonLAW, MoDIL, MoGUL, HYDRA
- Multi-genome callers (pooled) 
 - GenomeSTRiP, MoGUL, CommonLAW

BreakDancer



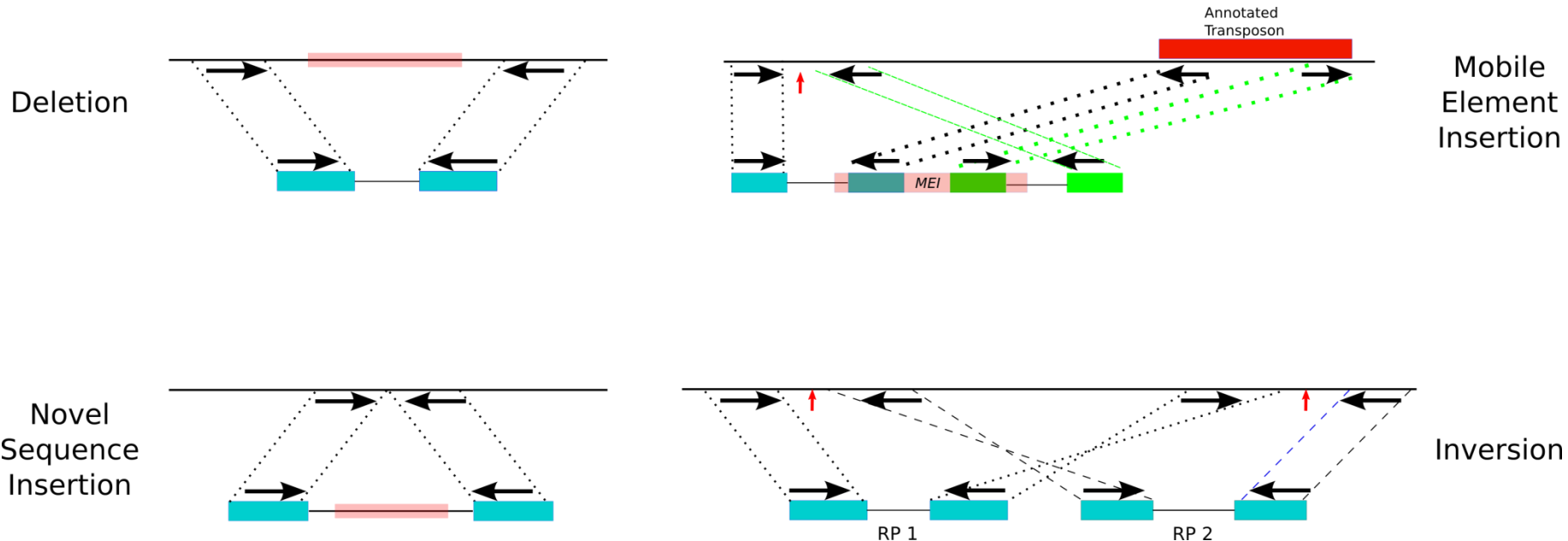
- Unique mapping from MAQ/BWA, etc. 
- Two versions:
 - BreakDancerMax
 - >100bp
 - BreakDancerMini
 - 10 – 100 bp

BreakDancerMax

- Unique mapping only; filter low MAPQ
- Classify inserts as:
 - Normal, deletion, insertion, inversion, intra-translocation, inter-translocation
 - If not “normal”, name as ARP (anomalous read pair)
- Call SV if at least 2 ARPs are at the same location 
- Assign confidence score

VariationHunter

- **VariationHunter-SC: Maximum parsimony approach; using all **discordant** map locations; finds an optimal set of SVs through a combinatorial algorithm based on *set-cover***
- *VariationHunter-Pr: Probabilistic version; tries to maximize the probability score of detected SVs*




VariationHunter (Steps)

- Uses multiple mappings from mrFast/mrsFast
 - Or convert BAM/SAM files into DIVET format
- Calculate the min/max fragment size
- Clustering Step : Cluster all the discordant reads into groups of consistent reads supporting an SV
- Selection: Among the created Clusters picks the minimum set which covers all (or most of the reads). Using *set cover* algorithm.
- Very similar to *HYDRA* method.

Trio based SV calling

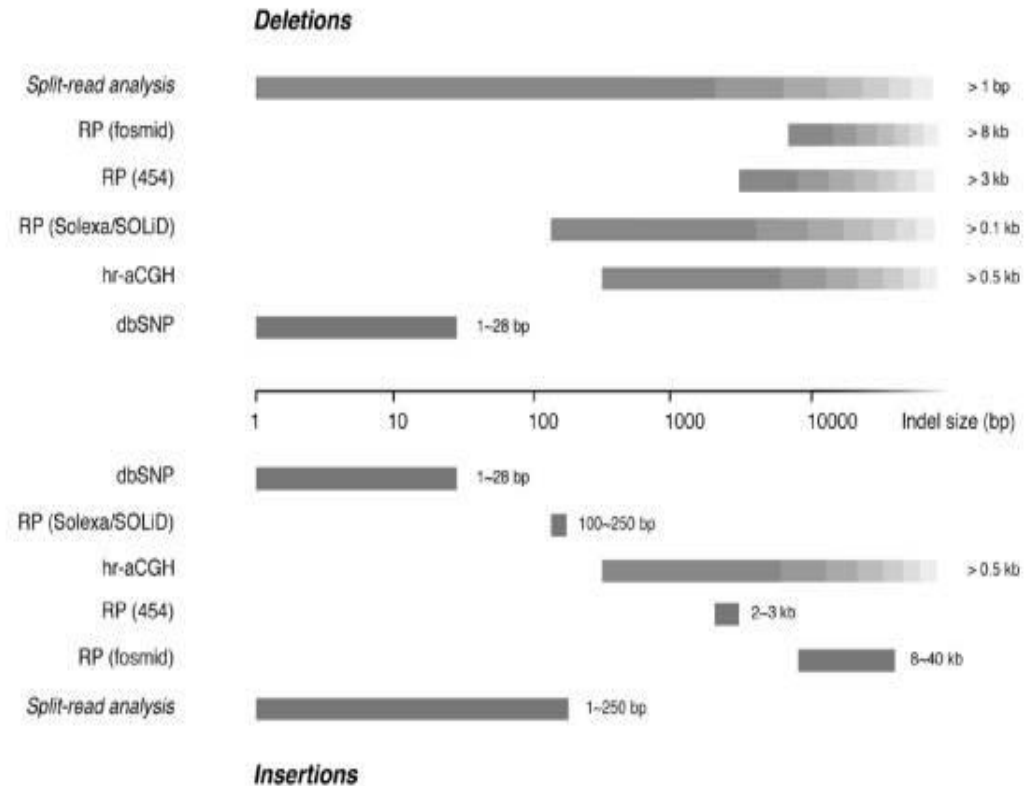


- Trio analysis for SV discovery is becoming very important.
- Calling de novo SVs is still a challenge (high FDR).
- Trio based methods use the mappings of the whole family simultaneously to predict event (including de novo SV)
- *CommonLaw* and *GenomeStrip* are two example of such methods. 

SPLIT READ

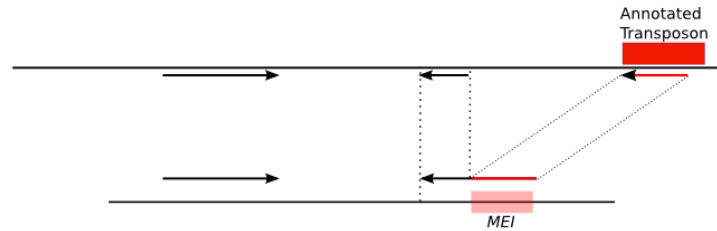
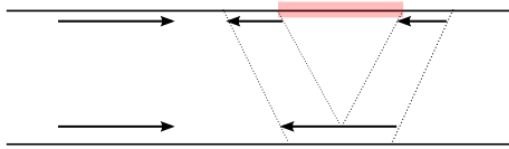
Split Read Approach Assumption

- The reads which span the breakpoint of the SV will not map as a whole but will be broken (split) in to two pieces.



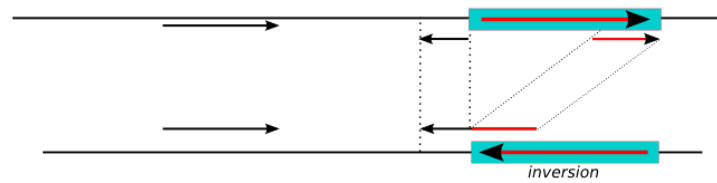
Split Read analysis

Deletion



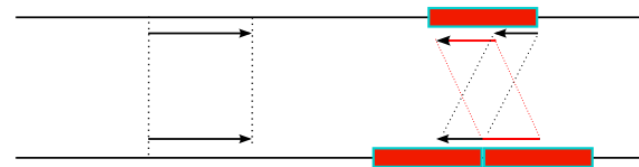
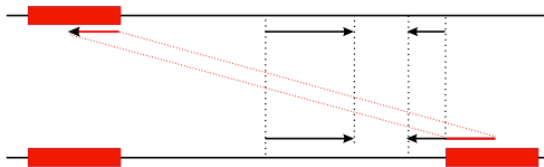
Mobile
Element
Insertion

Novel
Sequence
Insertion



Inversion

Interspersed
Duplication

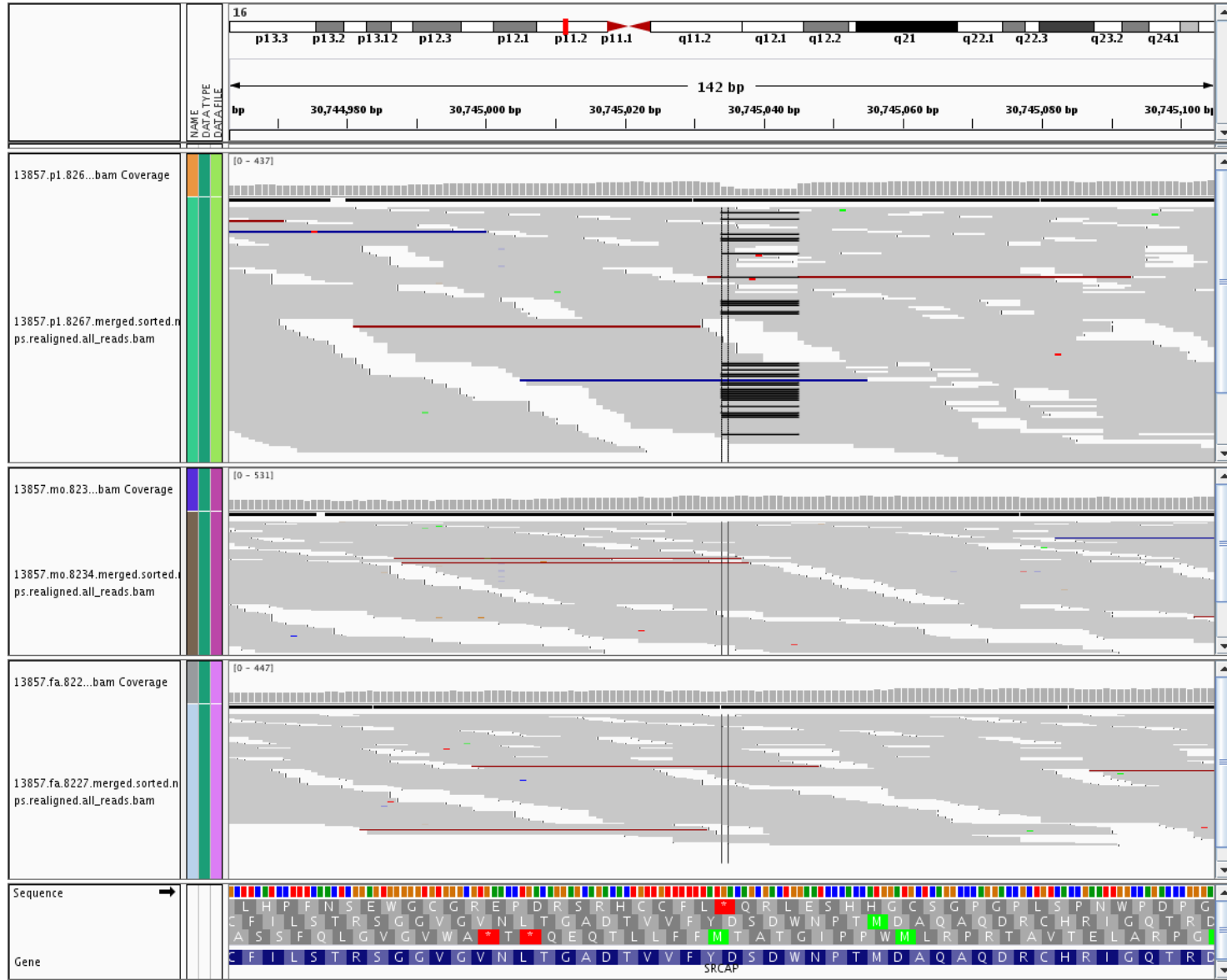


Tandem
Duplication

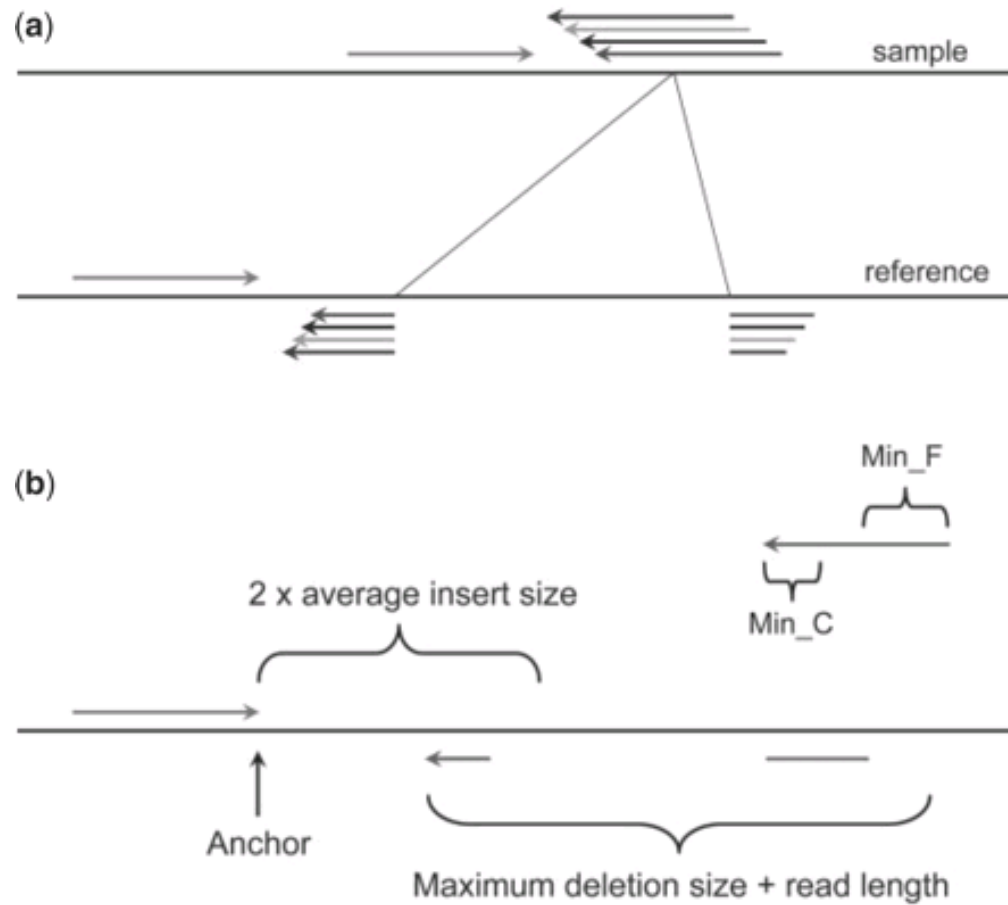
Split Read based algorithms

- Unique mapping:
 - Pindel (Ye et al. Bioinformatics, 2009)
 - SRiC (for the 454 platform; Zhang et al., BMC Bioinformatics, 2011)
- Multiple mapping:
 - SPLITREAD (Karakoc et al., Nature Methods, 2012)
- Specialized for RNA alternative splicing:
 - TopHat (Trapnell et al., Bioinformatics, 2009)

De novo deletion in SRCAP in Exome (GATK and Pindel)



Pindel: pattern growth approach




“Two heads are better than one”

MULTI SIGNATURE

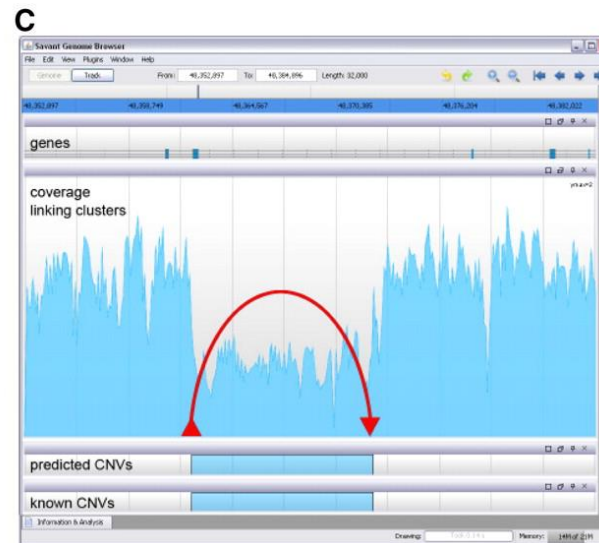
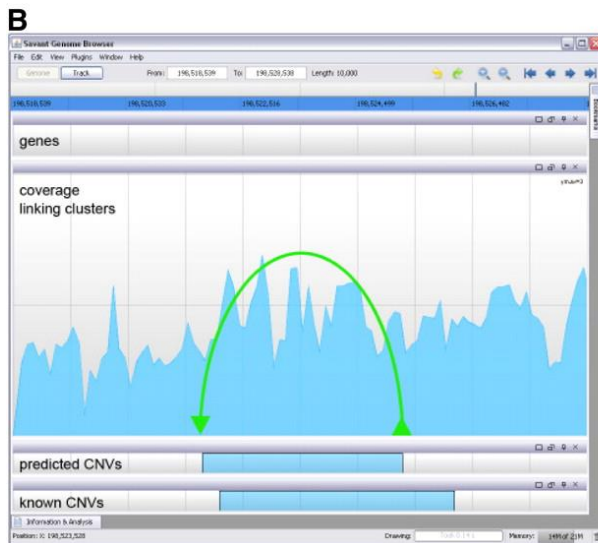
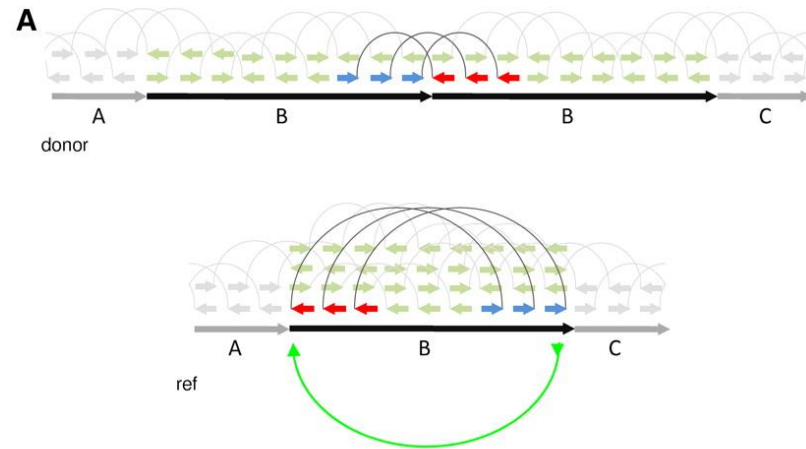
Combined Approach Assumption

- Some newer tools use two or more of the Read Depth, Read Pair or Split Read signatures to predict SV.
- There are biases which effect one signature more than others and using a combined approach we can reduce the False Discovery.

Multiple signature algorithms

- SPANNER (Stewart et al., unpublished) 
 - Find candidates with RP
 - Filter with RD
- Genome STRiP (Handsaker et al., Nat Genet, 2011)
 - Discovery: as above; also integrate **multiple genomes** in a population
 - Genotyping also includes SR
- CNVer (Medvedev et al., Genome Res, 2010)
 - Build a graph with RP; edge weights by RD
 - Solve minimum-cost-flow

CNVer

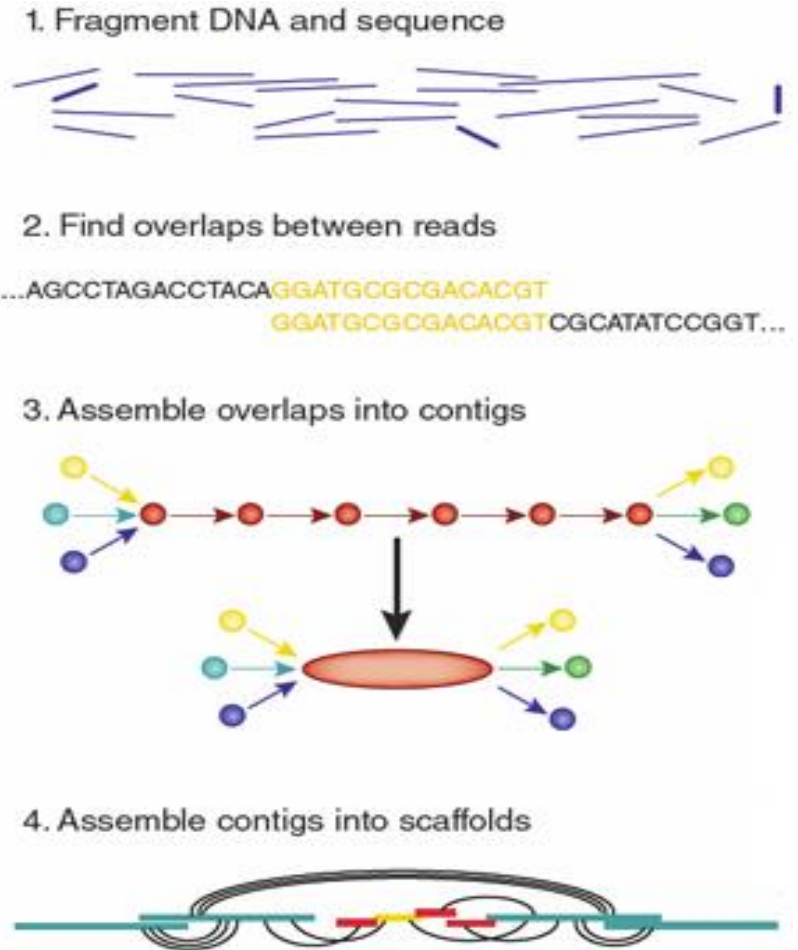




ASSEMBLY

Assembly

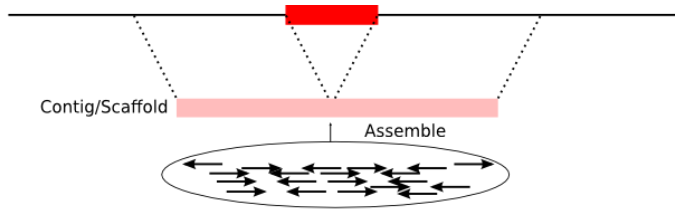
- Variant discovery using assembly approaches:
 - In assembly based approaches we first try to create the original donor genome using short reads (assembly step)
 - Compare the assembled Genome against the reference or other assembled genomes to predict variants



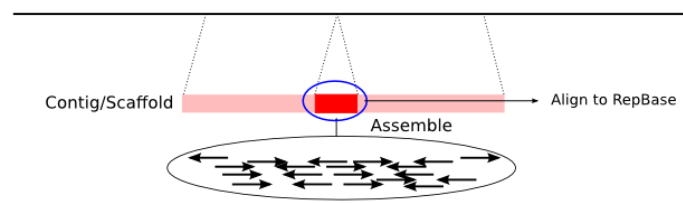
Genome assembly stitches together a genome from short sequenced pieces of DNA.

Assembly analysis

Deletion

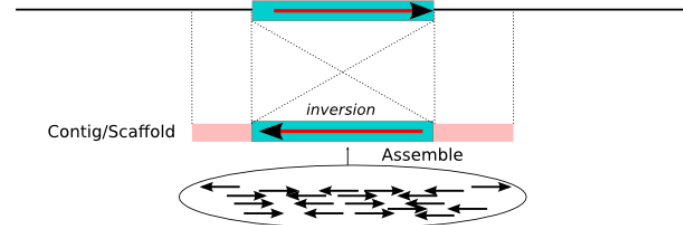
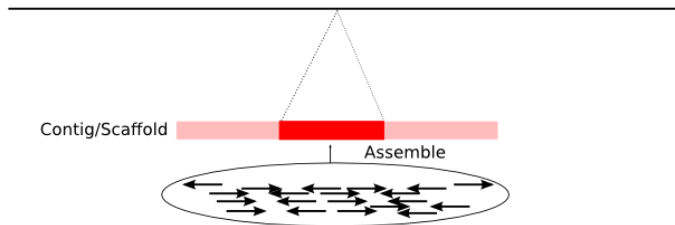


Mobile Element Insertion



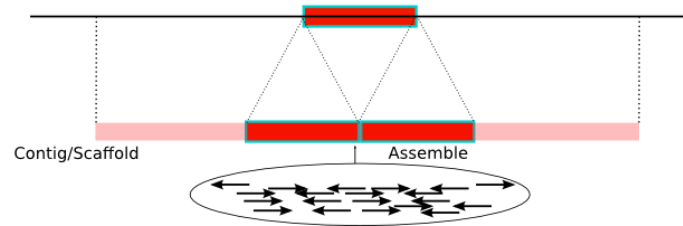
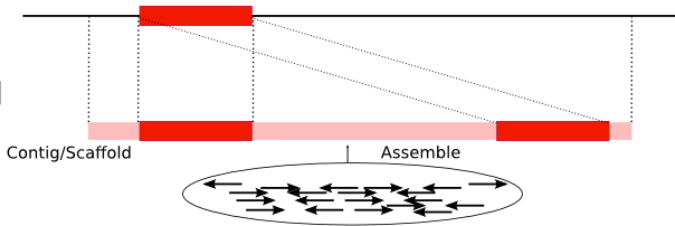
Mobile
Element
Insertion

Novel
Sequence
Insertion



Inversion

Interspersed
Duplication



Tandem
Duplication

Assembly approaches



Overlap-layout-consensus:

- 1.1) Overlap step: Create the *overlap* of the reads by doing a pairwise alignment.
- 1.2) Layout step: Create a graph based on the overlaps. Nodes with “significant” overlap are connected
- 1.3) Consensus: Create the assembly sequence by traversing the graph. Each node should be traversed exactly once.

This formulation was used in popular assemblers such as *Phrap* and *Celera* assembler.

The problem is NP-hard by simple reduction from

Hamiltonian Path problem. 

de bruijn graph assembly

- 1) Break the reads into k bp, where each will be a node in the graph.
- 2) Connect two nodes if they match $k-1$ bp
- 3) Find a Eulerian Path in this graph (a path which covers every edge exactly once)

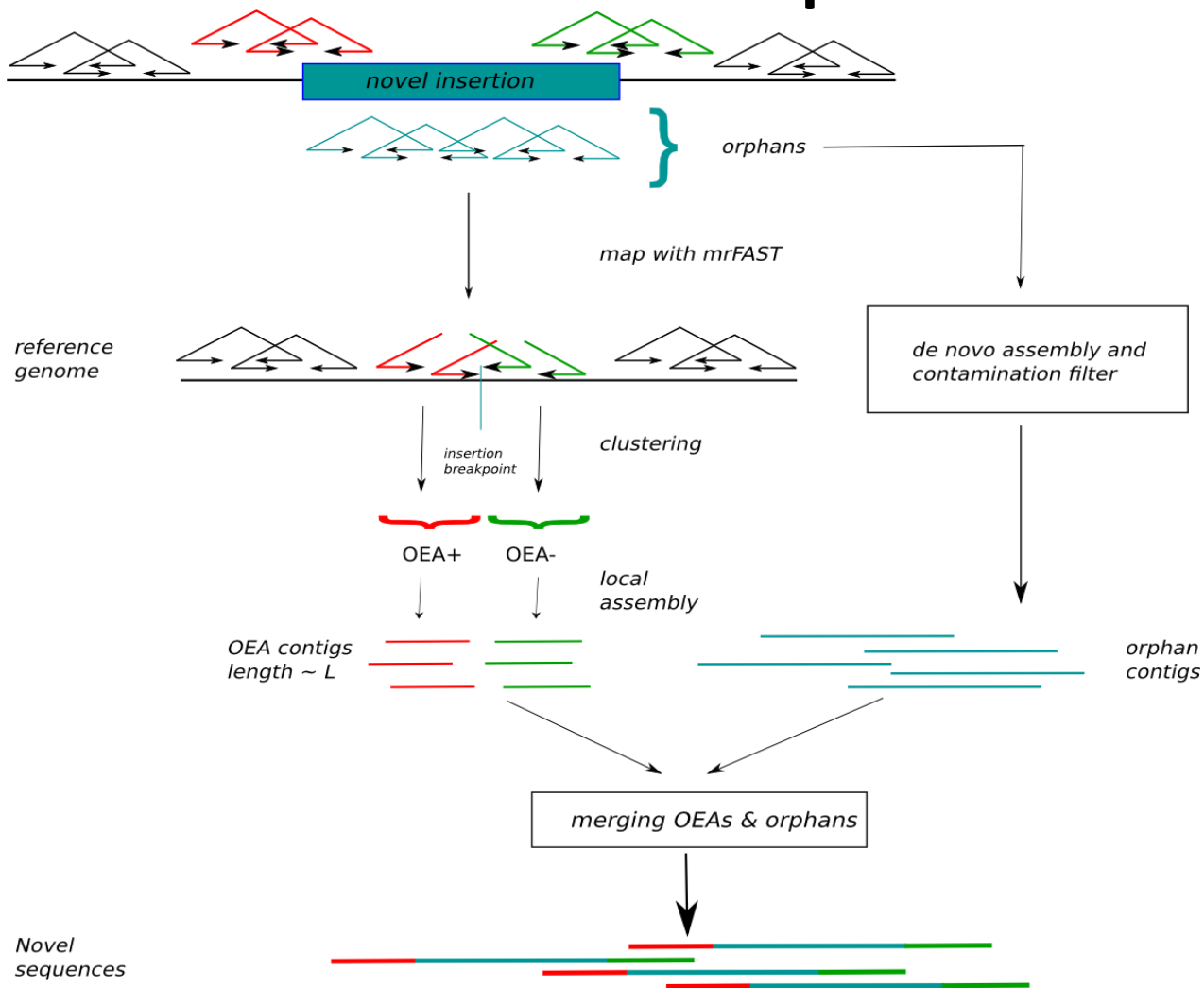
This formulation was used in newer assemblers such as Euler, Abyss, Velvet and Cortex.

The Eulerian path problem is easy to solve.

Assembly analysis

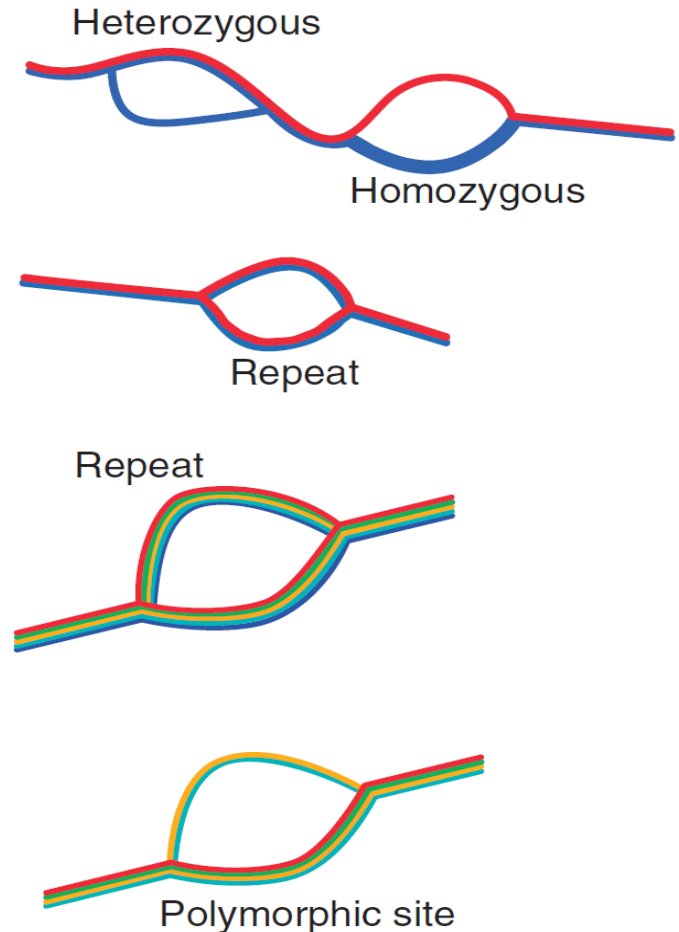
- Collect all reads; and assemble into contigs/scaffolds using:
 - Velvet, EULER, ABySS, Cortex, SOAPdenovo, ALLPATHS-LG, etc.
- Align to reference, and find SV
- SV-specific framework:
 - *NovelSeq: Going through the trash that the mapper left*

NovelSeq



Cortex Method

- Tries to predict variants between given set of genomes sequenced.
- Builds a multi-color de bruijn graph from the given inputs and calls the unicolor paths as a variant.

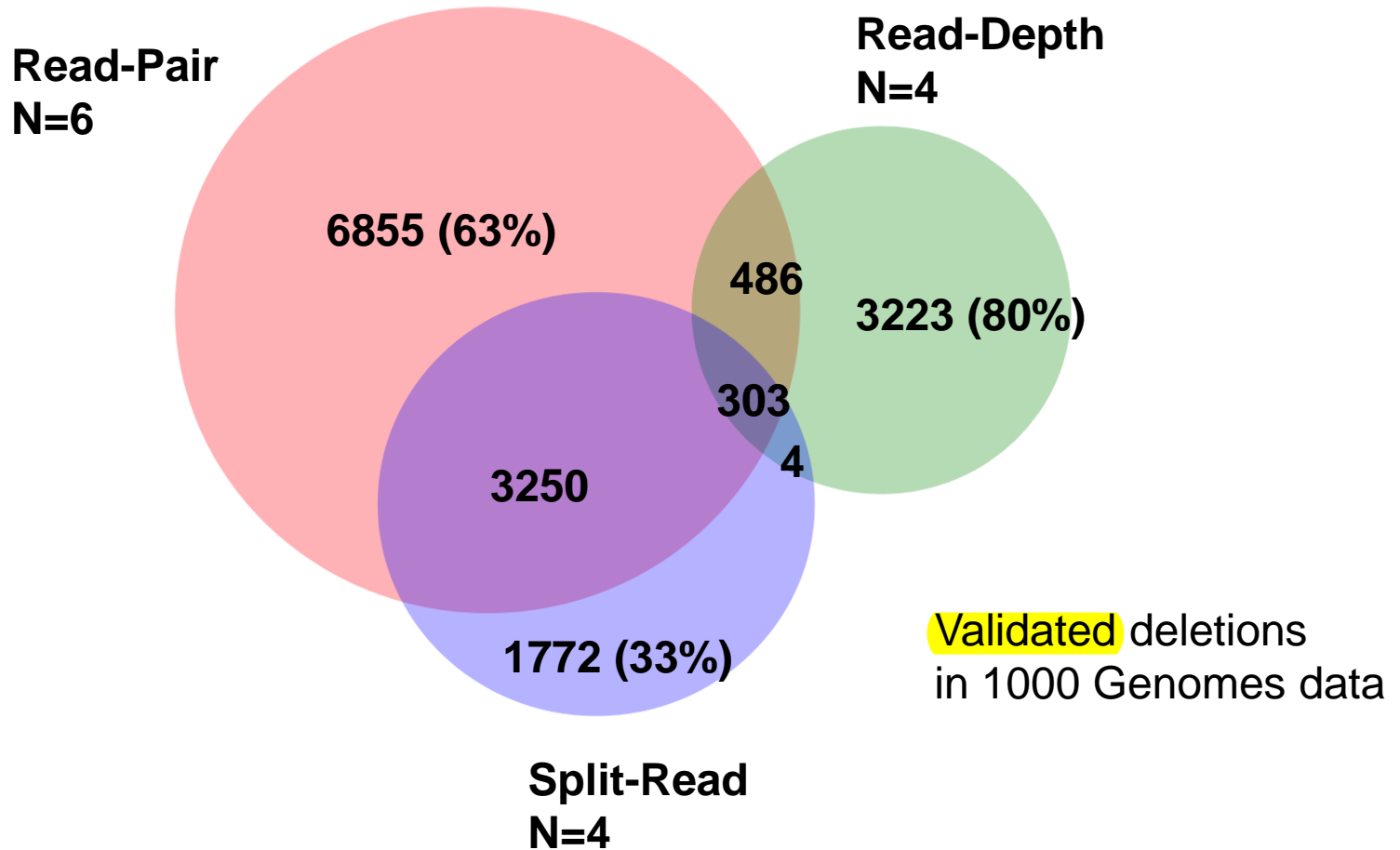


Cortex Method


- It can call SNPs, Indels and SV between different genomes provided.
- It is also capable to use the reference genome as one of the input samples and predict variants against it.
- High memory usage in comparison to previous methods: around 80G of RAM for one sample.

CONCLUSION & EXERCISE

No method is comprehensive




Open problems

- Identify ***inversions*** and ***translocations*** 
- Discover SVs in repeat- and duplication-rich regions
- Accurate & comprehensive detection of CNVs with a *single* algorithm
 - High sensitivity
 - High specificity

CommonLaw

EXERCISE

Exercise

- mrsFast mappings for chr22 of NA12878 trio is provided. 
- Run the CommonLaw method to find the SVs in the trio.
- Check to see which calls are supported by read depth of platinum genome.

Exercise

- Files you have/need:
 - VH (executable – for clustering)
 - multiInd_SetCover (executable – for picking SVs)
 - libFile (information about the libraries)
 - Mapping files for chr22 (*.vh.new)
 - initInfo, chr22, Hg19.Satellite, hg19_Gap.Table.USCS.Clean,
- In file commands the two main commands to run CL are provided.
 - 1) VH for clustering
 - 2) multiInd_SetCover for picking SV

Step 1

- `qlogin -q login.q -l h_vmem=10G`
- Create the *libInfo* file in your output folder. This file as the library name, individual name, location of the discordant mappings (in DIVET format), min insert size, max insert size and read length.

6

```
g1k-sc-NA12892-CEU-1 NA12892 ../day2/session3/vh_data/g1k_sc_NA12892_CEU_1.DIVET.vh.new.chr22 40 254 36
g1k-sc-NA12892-CEU-2 NA12892 ../day2/session3/vh_data/g1k_sc_NA12892_CEU_2.DIVET.vh.new.chr22 0 284 36
g1k-sc-NA12891-CEU-1 NA12891 ../day2/session3/vh_data/g1k-sc-NA12891-CEU-1.DIVET.vh.new.chr22 44 213 36
g1k-sc-NA12891-CEU-2.36bp NA12891 ../day2/session3/vh_data/g1k-sc-NA12891-CEU-2.36bp.DIVET.vh.new.chr22 0 273 36
g1k-sc-NA12878-CEU-1 NA12878 ../day2/session3/vh_data/g1k_sc_NA12878_CEU_1.DIVET.vh.new.chr22 93 184 36
g1k-sc-NA12878-CEU-2 NA12878 ../day2/session3/vh_data/g1k_sc_NA12878_CEU_2.DIVET.vh.new.chr22 48 309 36
```

Step 2 and 3

- Run the clustering step:

```
VH -i ../day2/session3/vh_data/Hg19_NecessaryFiles/initInfo -c  
../day2/session3/vh_data/Hg19_NecessaryFiles/chr22 -l libFile -t readName -o NA12878.trio.Clus -p 0.005 -x 100  
-g ../day2/session3/vh_data/Hg19_NecessaryFiles/hg19_Gap.Table.USCS.Clean -r  
../day2/session3/vh_data/Hg19_NecessaryFiles/Hg19.Satellite
```

- Run the selection step:

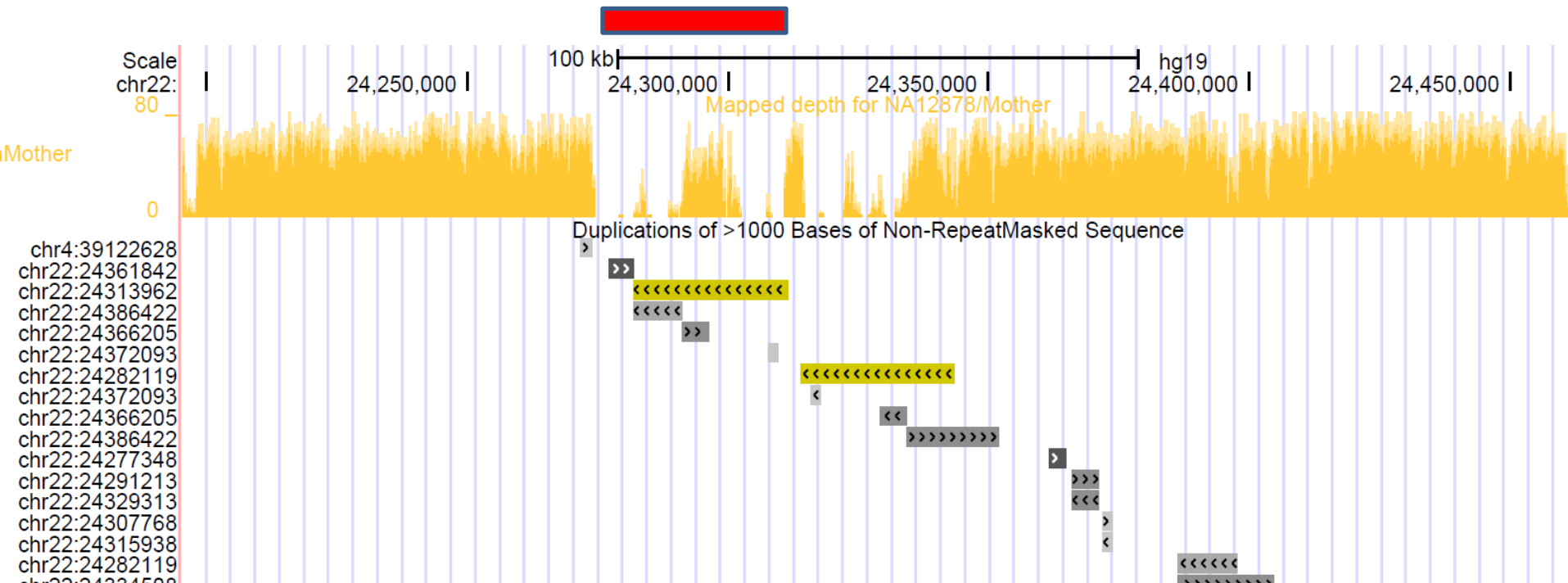
```
multiInd_SetCover -l libFile -r readName -c NA12878.trio.Clus -o NA12878.trio.Clus.SV -t 1000
```

- Deletions:

```
grep Svtype:D NA12878.trio.Clus.SV
```

Exercise

- There is predicted one deletion of size $> 10\text{kbp}$ with high support (>10 paired-end reads in NA12878).
Chr22: 24274148-24311299 (This deletion is validated by 1000G)



Exercise

- We can also visualize paired-end reads supporting same deletion.

