Variation detection based on second generation sequencing data

Xin LIU
Department of Science and Technology, BGI
liuxin@genomics.org.cn
2013.11.21
• Summary of sequencing techniques
• Data quality assessing and filtering
• Mapping the short reads
• Detection of SNPs
• Detection of SVs
• Detection of CNVs
Sequencing technologies

- Illumina
- SOLiD
- 454
- PacBio
- Complete Genomics
- Helicos
- Ion Torrent
- Oxford Nanopore
- Sanger
Illumina sequencing

Technology Spotlight: Illumina® Sequencing

Several samples can be loaded onto the eight-lane flow cell for simultaneous analysis on an Illumina Sequencing System.

Randomly fragment genomic DNA and ligate adapters.

Dense lawn of primers

Add unlabeled nucleotides and enzyme to initiate solid-phase bridge amplification.

Clusters

The first sequencing cycle begins by adding four labeled reversible terminators, primers, and DNA polymerase.

Several million dense clusters of double-stranded DNA are generated in each channel of the flow cell.
Ion Torrent sequencing

http://www.lifetechnologies.com
### Sequencing techniques

<table>
<thead>
<tr>
<th>Platform</th>
<th>Illumina MiSeq</th>
<th>Ion Torrent PGM</th>
<th>PacBio RS</th>
<th>Illumina GAIIx</th>
<th>Illumina HiSeq 2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instrument Cost*</td>
<td>$128 K</td>
<td>$80 K**</td>
<td>$695 K</td>
<td>$256 K</td>
<td>$654 K</td>
</tr>
<tr>
<td>Sequence yield per run</td>
<td>1.5-2Gb</td>
<td>20-50 Mb on 314 chip, 100-200 Mb on 316 chip, 1Gb on 318 chip</td>
<td>100 Mb</td>
<td>30Gb</td>
<td>600Gb</td>
</tr>
<tr>
<td>Sequencing cost per Gb*</td>
<td>$502</td>
<td>$1000 (318 chip)</td>
<td>$2000</td>
<td>$148</td>
<td>$41</td>
</tr>
<tr>
<td>Run Time</td>
<td>27 hours***</td>
<td>2 hours</td>
<td>2 hours</td>
<td>10 days</td>
<td>11 days</td>
</tr>
<tr>
<td>Reported Accuracy</td>
<td>Mostly &gt; Q30</td>
<td>Mostly Q20</td>
<td>&lt;Q10</td>
<td>Mostly &gt; Q30</td>
<td>Mostly &gt; Q30</td>
</tr>
<tr>
<td>Observed Raw Error Rate</td>
<td>0.80 %</td>
<td>1.71 %</td>
<td>12.86 %</td>
<td>0.76 %</td>
<td>0.26 %</td>
</tr>
<tr>
<td>Read length</td>
<td>up to 150 bases</td>
<td>~200 bases</td>
<td>Average 1500 bases**** (C1 chemistry)</td>
<td>up to 150 bases</td>
<td>up to 150 bases</td>
</tr>
<tr>
<td>Paired reads</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Insert size</td>
<td>up to 700 bases</td>
<td>up to 250 bases</td>
<td>up to 10 kb</td>
<td>up to 700 bases</td>
<td>up to 700 bases</td>
</tr>
<tr>
<td>Typical DNA requirements</td>
<td>50-1000 ng</td>
<td>100-1000 ng</td>
<td>~1 µg</td>
<td>50-1000 ng</td>
<td>50-1000 ng</td>
</tr>
</tbody>
</table>

Different genomic variations

Genome size

Karyotypes

DNA sequences

Chromosome rearrangements

Structural variations

Indels

SNPs
Detection of variations
Beginning, mapping/alignment

- Find the sequenced read’s placement in reference genome
- Calculate the coverage and depth distribution of the sequenced reads
- Sequencing quality evaluation
- Important for variation detection
Previous mapping tools

Needleman-Wunsch
- Global alignment algorithm
- An example: align COELACANTH and PELICAN
- Scoring scheme: +1 if letters match, -1 for mismatches, -1 for gaps

Smith-Waterman
- Modified to do local alignment

BLAST
- Three heuristic layers: seeding, extension, and evaluation
- Seeding – identify where to start alignment
- Extension – extending alignment from seeds
- Evaluation – Determine which alignments are statistically significant
• Differences between traditional and next-generation sequencing technology
  – reads length
  – data capacity
• Algorithm change to meet the data characteristics of the sequencing technology
  – traditional aligner: global or local alignment; scoring matrix; dynamic programming and trace-back
  – Next-Gen aligner: Indexing & Bitwise operation
• Does blastall/blat still work?
• Short Oligonucleotide Alignment/Analysis Package
Index the genome

### a

**Input**

GATATACACA

Repeated segments

**All rotations**

- GATATACACA
- AGATATACAC
- CAGATATAC
- ACAGATATAC
- CACAGATATA
- ACACAGATAT
- TACACAGATA
- ATACACAGAT
- TATACACAGA
- ATATACACAG

**Sorted rotations**

- ACACAGATAT
- ACAGATATAC
- CAGATATAC
- ATACACAGAT
- ATATACACAG
- CACAGATATA
- CAGATATA
- GATATACACA
- TACACAGATA
- TATACACAGA

**Output**

TCCTGAAAAA

More compressible

### b

**Suffix array**

<table>
<thead>
<tr>
<th>Suffix</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10</td>
</tr>
<tr>
<td>CA</td>
<td>9</td>
</tr>
<tr>
<td>ACA</td>
<td>8</td>
</tr>
<tr>
<td>CACA</td>
<td>7</td>
</tr>
<tr>
<td>ACACA</td>
<td>6</td>
</tr>
<tr>
<td>TACACA</td>
<td>5</td>
</tr>
<tr>
<td>ATACAC</td>
<td>4</td>
</tr>
<tr>
<td>TATACAC</td>
<td>3</td>
</tr>
<tr>
<td>ATATACAC</td>
<td>2</td>
</tr>
<tr>
<td>GATATACACA</td>
<td>1</td>
</tr>
</tbody>
</table>

**Query**

CAACA

**Sort**

<table>
<thead>
<tr>
<th>Suffix</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10</td>
</tr>
<tr>
<td>ACA</td>
<td>8</td>
</tr>
<tr>
<td>ACACA</td>
<td>6</td>
</tr>
<tr>
<td>ATACACA</td>
<td>4</td>
</tr>
<tr>
<td>TATACACACA</td>
<td>2</td>
</tr>
<tr>
<td>CA</td>
<td>9</td>
</tr>
<tr>
<td>CACA</td>
<td>7</td>
</tr>
<tr>
<td>GATATACACA</td>
<td>1</td>
</tr>
<tr>
<td>TACACA</td>
<td>5</td>
</tr>
<tr>
<td>TATACACACA</td>
<td>3</td>
</tr>
</tbody>
</table>
Algorithm of SOAP

- Comparing to BLAST, BLAT, short reads aligner applied looking-up method.
- Index of the reference were made in order to help the process of looking-up.
- Seed were first looked up by using index and then sequences were extended.
Features of SOAP

- Fast and efficient
- Mode of pair-end mapping
- Permit gaps within alignments
- Trim of reads permitted

<table>
<thead>
<tr>
<th>Program</th>
<th>Time consumed (s)</th>
<th>Reads aligned (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>blastn (-F F -W 11)</td>
<td>165780</td>
<td>85.47</td>
</tr>
<tr>
<td>blastn (-F F -W 15)</td>
<td>150660</td>
<td>84.66</td>
</tr>
<tr>
<td>Blat (-tileSize = 8)</td>
<td>22032</td>
<td>85.07</td>
</tr>
<tr>
<td>Eland</td>
<td>166</td>
<td>88.53</td>
</tr>
<tr>
<td>Maq</td>
<td>458</td>
<td>88.39</td>
</tr>
<tr>
<td>Soap</td>
<td>134</td>
<td>88.46</td>
</tr>
<tr>
<td>Soap iterative</td>
<td>161</td>
<td>90.9</td>
</tr>
<tr>
<td>Soap iterative + gapped</td>
<td>486</td>
<td>91.15</td>
</tr>
</tbody>
</table>
Algorithm of SOAP2

• Indexing
  – Split read into parts, which used to anchor the exact matching region in the reference, excluding much of the unwanted region.

• 2way-BWT (Burrows-Wheeler transform) provide a excellent solution for the computing complexity
  – Memory effective (~7G memory need for 3G genome).
  – Fast indexing (2 minutes to finish 1M 35bp single end alignment).

• Thread Parallel Computing
  – Make fully use of process and save time.

• Bitwise operation
  – Encode each base into 2 binary bits, and use exclusive-or to check if two bases are the same.
Flowchart of mapping

Pre-build index files

- Reference
- BWT Reference
- Suffix Array
- Common Prefixes
  Hash

Search queries

Branching Limited
(Mismatches)
BWA

- Burrows-Wheeler Alignment tool (BWA), the read alignment package that is based on backward search with Burrows–Wheeler Transform (BWT);
- Allowing mismatches and gaps;
- BWA is \( \sim 10-20 \times \) faster than MAQ, while achieving similar accuracy;
- BWA outputs alignment in the SAM format
- Variant calling can be achieved by SAMtools software package
## CPU time and RAM

<table>
<thead>
<tr>
<th>Aligners</th>
<th>CPU-TIME (min)</th>
<th>RAM (G)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>35</td>
<td>75</td>
</tr>
<tr>
<td>BWA</td>
<td>50</td>
<td>14</td>
</tr>
<tr>
<td>Bowtie</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>BFAST</td>
<td>300</td>
<td>360</td>
</tr>
<tr>
<td>MAQ</td>
<td>300</td>
<td>180</td>
</tr>
<tr>
<td>Mosaik</td>
<td>122</td>
<td>50</td>
</tr>
<tr>
<td>mrFAST</td>
<td>900</td>
<td>284</td>
</tr>
<tr>
<td>Novoalign</td>
<td>149</td>
<td>18</td>
</tr>
<tr>
<td>SOAP2</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Stampy</td>
<td>206</td>
<td>109</td>
</tr>
</tbody>
</table>
Sensitivity of mapping

BFAST
MAQ
Bowtie
mrFAST
SOAP2
BWA
Mosaik
Stampy
Novoalign

(%) of Reads

Correct  Incorrect  Unmap
Correctness of mapping

- MAQ
- BFAST
- Bowtie
- BWA
- SOAP2
- Stampy
- Novoalign
- mrFAST
- Mosaik

(%) of Reads

86 88 90 92 94 96 98 100

21
Alignment with insertion

- 35bp Reads' PPV(DEL)
- 75bp Reads' PPV(DEL)
- 90bp Reads' PPV(DEL)
- 101bp Reads' PPV(DEL)

Graphs showing the performance of various alignment tools with different read lengths (35, 75, 90, 101 bp) and deletion lengths on the x-axis, and PPV (%) on the y-axis. The tools compared include BFAST, BWA, NAQ, Mosaic, Novoalign, mrFAST, SOAP2, and Stampy.
Summary of mapping

• Short reads mapping need specific aligners
• Many aligners are available and there are different features.
• No best aligner exists, and most of them are acceptable.
• Mapping is important in variation detection.
Importance of SNPs

• Well-studied variation
• Better representing demographic history
• Method of detection is relatively mature
• Provides more information for follow-up studies
• Detect SNPs in individuals
• Detect SNPs in population
SOAPsnp to detect SNPs

• SOAPsnp was developed for consensus calling and SNP detection based on the Solexa sequencing technology.
• SOAPsnp takes Bayes’s theorem as statistic model for SNP calling, it considers:
  – Sequencing quality
  – Likelihood calculation based on observed data
  – Experiment factors
  – Prior probability
  – Alignment uniqueness and accuracy
  – Using dbSNP as prior probability
SOAPsnp

1. Sequencing reads
2. Map reads onto reference genome
3. Recalibrate sequencing quality score
4. Calculate likelihood of each genotype
5. Inferred genotype via Bayes’ theorem

Reference: G

10 Possible genotypes for diploid

Observed data (base, quality)
G24, G25, G18, G20, G25, A15, A18, A30, A25, T10

Possible called consensus

A  A  C  T  A  G  C  T  C  G  T  G  G
Sequencing errors

- SNP identification can be inferred by counting mismatch numbers.
- But, sequencing quality is important for distinguishing sequencing error from SNP, especially for Solexa sequencing.
Statistic model

\[ P(T_i | D) = \frac{P(T_i)P(D | T_i)}{\sum_{x=1}^{S} P(T_x)P(D | T_x)} \]

- D: is the observed data in alignment.
- Prior\((g)\): prior probability of a given genotypes
- \(P(D | x)\): conditional probability to get the observed data \(D\) of a given genotype

Diploid Ti contain 10 types: AA, CC, GG, TT, AC, AG, AT, CG, CT, GT;

\[
P((\text{Base, quality}) | Ti) = \frac{P((\text{Base, quality}) | Ti_1)}{2} + \frac{P((\text{Base, quality}) | Ti_2)}{2}
\]

\[
P(D | Ti) = \prod P((\text{Base, quality}) | Ti)
\]

### Table 1. Prior probability of genotypes of a diploid genome

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3.33 \times 10^{-4}</td>
<td>1.11 \times 10^{-7}</td>
<td>6.67 \times 10^{-4}</td>
<td>1.11 \times 10^{-7}</td>
</tr>
<tr>
<td>C</td>
<td>8.33 \times 10^{-5}</td>
<td>1.67 \times 10^{-4}</td>
<td>2.78 \times 10^{-8}</td>
<td>8.33 \times 10^{-5}</td>
</tr>
<tr>
<td>G</td>
<td>0.9985</td>
<td>1.67 \times 10^{-4}</td>
<td>1.67 \times 10^{-4}</td>
<td>8.33 \times 10^{-5}</td>
</tr>
<tr>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Assuming that the reference allele is G, the homozygous SNP rate is 0.0005, the heterozygous SNP rate is 0.001, and the ratio of transitions versus transversions is 4.
Filtering SNPs

• By comparing the reference allele and the consensus genotype with maximum Bayesian likelihood, decisions of SNP status are made.

• But this is only the candidate SNPs, the accuracy is not reliable, especially when the mapped reads depth is low.

• In addition, we used some other measures to get confident SNPs, such as the minimum supporting reads number for each allele, exclude SNP predictions on repeat regions, and the rank-sum test for heterozygous SNP.
Other software to identify SNPs

- MAQ provides modules to identify SNPs. Error rates were calculated at each position thus SNPs were identified.

- Samtools, using sam/bam files, identifies SNPs in individual or populations.
Detect SNPs at different depth

In population studies, sequencing depth of each individual is always low. Then, how can we detect SNPs?
Detect SNPs in population

- Sequencing in population
  - Several individuals sequenced
  - Sequencing depth of each individual is relatively low (0.1-20X).
  - Total depth is high, several hundred times.
## Detect SNPs in population

<table>
<thead>
<tr>
<th>Sequencing depth</th>
<th>Genome coverage ratio</th>
<th>Identified SNP ratio</th>
<th>Study purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low (1-3X)</td>
<td>50%-80%</td>
<td>30%-50%</td>
<td>Rough population survey, infer population structure, phylogeny, and selection.</td>
</tr>
<tr>
<td>Middle (6-10X)</td>
<td>90%-99%</td>
<td>70%-90%</td>
<td>Whole population sequencing, suitable for further applications, such as molecular inbreeding, and functional genomics.</td>
</tr>
<tr>
<td>High (20-40X)</td>
<td>99.9%~100%</td>
<td>95%-99%</td>
<td>Complete map (de novo assembly) for each subspecies, line, or individual, suitable for all kinds of future applications.</td>
</tr>
</tbody>
</table>
• To detect SNPs in population (applied in silkworm paper)

Mapping
• Using SOAP to align sequences from each individual to the reference.

SOAPsnp
• Using SOAPsnp to determine the likelihood of genotypes at each position in each individual.

GLFmulti
• Integrate the likelihood of each individual at each position, then apply MLE to estimate the allele frequency.
• Frequency at each site with the maximum likelihood is given.
• Copy number, sequencing depth, quality score and minor allele count are integrated into one score.
• SNPs were confidential, but SNPs at low frequency were underestimated.
To detect SNPs in population (applied in Tibetan paper)

**Mapping**
- Using SOAP to align sequences from each individual to the reference.

**SOAPsnp**
- Using SOAPsnp to determine the likelihood of genotypes at each position in each individual.

**realSFS**
- Calculate the likelihood of allele frequency at each position.
1. Likelihood of different allele frequency.

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual1</td>
<td>P0</td>
<td>P1</td>
<td>P2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Individual2</td>
<td>P0*P0(2)</td>
<td>P1<em>P0(2)+P0</em>P1(2)</td>
<td>P2<em>P0(2)+P0</em>P2(0)+P1*P1(2)</td>
<td>P1<em>P2(2)+P2</em>P1(0)</td>
<td>P2*P2(2)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

2. Prior probability of different allele frequency.
• Detect SNPs at low depth of each individual (lowest 0.4X depth).
• Relatively higher false positive ratio.
Other software to detect SNPs

• GATK: a widely used software/pipeline to detect variations, especially for human population (http://www.broadinstitute.org/gsa/wiki/index.php/Main_Page)

• SHORE: a pipeline used in 1000 genome project of Arabidopsis (http://sourceforge.net/apps/mediawiki/shore/index.php?title=SHORE_Documentation)
The GenomeAnalysisToolkit (GATK) enabling rapid development of efficient and robust analysis tools

- Manages basic program infrastructure
- Libraries for accessing data in many formats and conversion to standard data structures
- Automatic threading, distributed computing, and other high-performance features
- Provides structured and efficient access to reads, reference bases, and metadata
- Analysis-specific calculation using data presented to it by traversal engine
- Relies on the engine to manage the data interaction to focus on analysis calculation
SHORE is a data analysis and management application for short DNA/RNA reads produced by the various contemporary sequencing platforms.

SHORE is designed to support different sequencing applications including genomic re-sequencing, ChIP-Seq, mRNA-Seq, sRNA-Seq and BS-seq.

SHORE was developed for applications in *Arabidopsis thaliana* but has been successfully used with other genomes, including human, mouse, *D. melanogaster*, *C. elegans*, maize and several bacterial genomes.
Summary of SNP detection

• Different methods can be applied in SNP detection in single individual.
• The main problem to cope with is the sequencing errors which would result in false positive.
• The variation calling would also depend a lot on the mapping result.
• Experimental validation is necessary.
• Detect SNPs in individual require higher depth.
• Detect SNPs in population can detect SNPs at lower individual depth.
• Statistic method is usually applied in SNP calling to prevent influence of sequencing errors.
• Different statistic models show different detection power.
Detection of indels

- After mapping, if the mapping permits gaps, those alignments with gaps can be the candidate for indels.
- In SOAPindel sequencing quality and mapping result were combined to deduce the probability of being an indel.
Detection of indel by SOAPindel

Reference

Mapped reads

Homozygous deletion

Heterozygous deletion

Homozygous insertion

Heterozygous insertion
Other methods to detect indels

• **Dindel**: program for calling small indels from short-read sequence data
  – Extracts all indels from the read-alignments in the BAM file
  – Candidate InDels grouped into windows
  – For each window, Dindel will generate candidate haplotypes from the candidate indels and realign
  – Interpreting the output from Dindel
Detection of SVs

• Mainly there are three kinds of methods to detect structural variations:
  – Local *de novo* assembly
  – Pair end mapping
  – Split reads

• The SV detection based on assembly is believed to be more accurate.
Local *de novo* assembly

1. Assemble pair-end reads into scaffold.
2. Align the scaffold to reference genome.
Workflow of SOAPsv:
Complex structure variants can be detected:

Output format:

(#Type, scaffold, start, end, refChr, start, end, length, sequence)

Reference:

Pair end mapping

1. Map the reads to the reference genome.
2. Detect structure variants by discordant reads.
Workflow of BreakDancer:

(a) Workflow diagram:
- Paired-end reads → Mapping parameters → Mapping → Detection parameters → (i) Genomewide tally of anomalous read pairs → (ii) Search for anomalous regions
- Structural variants → (v) Compute confidence scores → (iv) Structural variation position, type, size and number of anomalous mapped read pairs → (iii) Identify interconnected clusters

(b) Diagram showing types of structural variants:
- Deletion
- Insertion
- Inversion
- Intrachromosomal translocation
- Interchromosomal translocation
Performance of BreakDancer

Output format:

(#chromosome, position, orientation, chromosome, position, orientation, type, length, score ...)

Reference:

1. Map the reads to reference genome.
2. Select those paired reads that mapped with indels or of which only one end can be mapped.
3. Uses the mapped reads to determine the anchor point on the reference genome and the direction of the unmapped reads.
Workflow of Pindel:

Alignment BAM file generated by BWA
- create BAM config file (recommended)
- BAM config file

Alignment BAM-file generated by other aligners
- Run bam2pindel (deprecated)
- Alignment SAM-file
- samtools
- sam2pindel.cpp
- Pindel text input file

Run pindel
- Deletions file (\_D)
- Short insertions file (\_SI)
- Long insertions file (\_SI)
- Inversions file (\_INV)
- Tandem duplications file (\_TD)
- Breakpoints file (\_BP)
SLINEs and LINEs can be detected by Pindel:

Output format:

(#index, type, length, insertion length, chrID, border of event, range of unclear breakpoint, support number)

Reference:

### Comparison

<table>
<thead>
<tr>
<th>Tools</th>
<th>SOAPsv</th>
<th>BreakDancer</th>
<th>Pindel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main detectable length range</td>
<td>1 bp-50 kbp</td>
<td>&gt;10bp</td>
<td>1bp-30kbp</td>
</tr>
</tbody>
</table>

Detectable SV types

<table>
<thead>
<tr>
<th></th>
<th>SOAPsv</th>
<th>BreakDancer</th>
<th>Pindel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insertions</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Deletions</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Inversions</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Complex</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Precision of breakpoints

<table>
<thead>
<tr>
<th></th>
<th>SOAPsv</th>
<th>BreakDancer</th>
<th>Pindel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Single base</td>
<td>A short ambiguous range</td>
<td>Single base</td>
</tr>
</tbody>
</table>

Genotypes of SV events

<table>
<thead>
<tr>
<th></th>
<th>SOAPsv</th>
<th>BreakDancer</th>
<th>Pindel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

False-positive rate in simulated data

<table>
<thead>
<tr>
<th></th>
<th>SOAPsv</th>
<th>BreakDancer</th>
<th>Pindel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.20%</td>
<td>9.1–10.3%</td>
<td>&lt;2%</td>
</tr>
</tbody>
</table>

False-negative rate in simulated data

<table>
<thead>
<tr>
<th></th>
<th>SOAPsv</th>
<th>BreakDancer</th>
<th>Pindel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>9.60%</td>
<td>26–32%</td>
<td>~20%</td>
</tr>
</tbody>
</table>

The CNV detection

- **Depth of Coverage (DOC):**
  - Number of reads in a region
  - Uniform depth distribution
  - Biased for GC etc.
- **Paired End Mapping (PEM):**
  - Proper pairing when mapping
  - Limit by insert sizes
  - Inversions/Translocations
- **Split Reads (SR):**
  - The unaligned reads
  - Pinpoint the location of CNV
- **Assembly based (AS)**

Zhao et al. BMC Bioinformatics 2013, 14(Suppl 11):S1
<table>
<thead>
<tr>
<th>Method</th>
<th>URL</th>
<th>Language</th>
<th>Input</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEM-based</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BreakDancer</td>
<td><a href="http://breakdancer.sourceforge.net/">http://breakdancer.sourceforge.net/</a></td>
<td>Perl, C++</td>
<td>Alignment files</td>
<td>Predicting insertions, deletions, inversions, inter- and intra-chromosomal translocations</td>
</tr>
<tr>
<td>PEMer</td>
<td><a href="http://svgersteinlab.org/permer/">http://svgersteinlab.org/permer/</a></td>
<td>Perl, Python</td>
<td>FASTA</td>
<td>Using simulation-based error models to call SVs</td>
</tr>
<tr>
<td>VariationHunter</td>
<td><a href="http://compbio.cs.sfu.ca/strvar.htm">http://compbio.cs.sfu.ca/strvar.htm</a></td>
<td>C</td>
<td>DIVETa</td>
<td>Detecting insertions, deletions and inversions</td>
</tr>
<tr>
<td>commonLAW</td>
<td><a href="http://compbio.cs.sfu.ca/strvar.htm">http://compbio.cs.sfu.ca/strvar.htm</a></td>
<td>C++</td>
<td>Alignment files</td>
<td>Aligning multiple samples simultaneously to gain accurate SVs using maximum parsimony model</td>
</tr>
<tr>
<td>GASV</td>
<td><a href="http://code.google.com/p/gasv/">http://code.google.com/p/gasv/</a></td>
<td>Java</td>
<td>BAM</td>
<td>A geometric approach for classification and comparison of structural variants</td>
</tr>
<tr>
<td>Spanner</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>Using PEM to detect tandem duplications</td>
</tr>
<tr>
<td>SR-based</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AGE</td>
<td><a href="http://svgersteinlab.org/age">http://svgersteinlab.org/age</a></td>
<td>C++</td>
<td>FASTA</td>
<td>A dynamic-programming algorithm using optimal alignments with gap excision to detect breakpoints</td>
</tr>
<tr>
<td>Pindel</td>
<td><a href="http://www.ebi.ac.uk/~eye/pindel/">http://www.ebi.ac.uk/~eye/pindel/</a></td>
<td>C++</td>
<td>BAM /FASTQ</td>
<td>Using a pattern growth approach to identify breakpoints of various SVs</td>
</tr>
<tr>
<td>SLOPE</td>
<td><a href="http://www-genepi.med.utah.edu/suppl/">http://www-genepi.med.utah.edu/suppl/</a></td>
<td>C++</td>
<td>SAM/MAQ /FASTQ</td>
<td>Locating SVs from targeted sequencing data</td>
</tr>
<tr>
<td>SRIC</td>
<td>N/A</td>
<td>N/A</td>
<td>BLAT output</td>
<td>CalibratingSV calling using realistic error models</td>
</tr>
<tr>
<td>AS-based</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Magnolya</td>
<td><a href="http://sourceforge.net/projects/magnolya/">http://sourceforge.net/projects/magnolya/</a></td>
<td>Python</td>
<td>FASTA</td>
<td>Calling CNV from co-assembled genomes and estimating copy number with Poisson mixture model</td>
</tr>
<tr>
<td>Cortex assembler</td>
<td><a href="http://cortexassembler.sourceforge.net/">http://cortexassembler.sourceforge.net/</a></td>
<td>C</td>
<td>FASTQ/FASTA</td>
<td>Using alignment of de novo assembled genome to build de Bruijn graph to detect SVs</td>
</tr>
<tr>
<td>TiGRA-SV</td>
<td><a href="http://gmt.genomewustl.edu/tigra-sv/">http://gmt.genomewustl.edu/tigra-sv/</a></td>
<td>C</td>
<td>SV calls + BAM</td>
<td>Local assembly of SVs using the iterative graph routing assembly (TIGRA) algorithm</td>
</tr>
</tbody>
</table>

### Method

<table>
<thead>
<tr>
<th>Method</th>
<th>URL</th>
<th>Language</th>
<th>Input</th>
<th>Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>NovelSeq</td>
<td><a href="http://compbio.cs.sfu.ca/strvar.htm">http://compbio.cs.sfu.ca/strvar.htm</a></td>
<td>C</td>
<td>FASTA/SAM</td>
<td>PEM + AS</td>
</tr>
<tr>
<td>HYDRA</td>
<td><a href="http://code.google.com/p/hydra-sv/">http://code.google.com/p/hydra-sv/</a></td>
<td>Python</td>
<td>Discordant paired-end mappings</td>
<td>PEM + AS</td>
</tr>
<tr>
<td>CNVer</td>
<td><a href="http://compbio.cs.toronto.edu/CNVer/">http://compbio.cs.toronto.edu/CNVer/</a></td>
<td>Perl, C++</td>
<td>BAM/aligned positions</td>
<td>PEM + RD</td>
</tr>
<tr>
<td>GASV Pro</td>
<td><a href="http://code.google.com/p/gasv/">http://code.google.com/p/gasv/</a></td>
<td>C++</td>
<td>BAM</td>
<td>PEM + RD</td>
</tr>
<tr>
<td>Genome STRIP</td>
<td><a href="http://www.broadinstitute.org/software/genomestrip/genome-strip">http://www.broadinstitute.org/software/genomestrip/genome-strip</a></td>
<td>Java, R</td>
<td>BAM</td>
<td>PEM + RD</td>
</tr>
<tr>
<td>SVDetect</td>
<td><a href="http://svdetect.sourceforge.net/">http://svdetect.sourceforge.net/</a></td>
<td>Perl</td>
<td>SAM/BAM/ELAND</td>
<td>PEM + RD</td>
</tr>
<tr>
<td>inGAP-sv</td>
<td><a href="http://ingap.sourceforge.net/">http://ingap.sourceforge.net/</a></td>
<td>Java</td>
<td>SAM</td>
<td>PEM + RD</td>
</tr>
<tr>
<td>SVseq</td>
<td><a href="http://www.engr.uconn.edu/~jlx08001/svseq.html">http://www.engr.uconn.edu/~jlx08001/svseq.html</a></td>
<td>C</td>
<td>FASTQ/BAM</td>
<td>PEM + SR</td>
</tr>
<tr>
<td>Nord et al.</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>RD + SR</td>
</tr>
</tbody>
</table>
Software to detect CNV

- CNVer [http://compbio.cs.toronto.edu/CNVer/](http://compbio.cs.toronto.edu/CNVer/)
Notes about CNV detection

• Repeat regions would have great impact.
• Mapping depth should be carefully inspected especially when the pair-end mapping was done.
• It is always difficult to give the actual copy numbers.
Summary of variation detection

• Different models are available in SNP calling of a single individual, and good methods should take care of both sequencing and mapping quality.
• Indels can be easily detected by interpreting the gapped alignment, and the accuracy do depend greatly on the mapping.
• Using assembly to find SVs is more accurate in practice than using pair-end information.
• Depth of coverage was applied in CNV detection.
Thanks!