Understanding
Next Generation Sequencing (NGS)
Technology

Robert Yu
September 2013
Working on our recent research projects using NGS data, I realized that I have to adopt more new tool kits, to learn more new concepts, new data processing techniques and new analysis methods. And thus this presentation.

Rather than a presentation of a completed study, it is more of the notes of my ongoing learning NGS technology.

Therefore, the content listed in the subsequent slides could be improperly organized and incomplete at all. However, I hope the presentation process could further clarify myself on the learning and gather your additions and suggestions to my more effective learning process.

Thank you.
• Overview
• DNA Sequencing
  ♦ Sanger sequencing
• NGS Technology
  ♦ Overview
  ♦ Resequencing
  ♦ De novo Sequencing
• NGS Data Processing
  ♦ Workflow using GATK
  ♦ Reads mapping → SAM/BAM files
  ♦ Variants Detection → VCF files
• SNP VCF
  ♦ SNP and Indel Calling
  ♦ SNP VCF Conversion to PLINK by PSEQ
  ♦ SNP analysis
• Summary

Ref:
Overview
Conventional low resolution genotyping
*e.g. microsatellite marker, low density SNP, etc.*

**Genomic DNA**
→ dense genotyping
→ better genetic studies

Whole genome sequencing or target (*regional*) sequencing

Technical challenges in large-scale of DNA sequencing and post-sequencing data processing
→ NGS and sequencing data analyses
Conventional DNA sequencing method

- **The classical chain-termination method** requires a single-stranded DNA template, a DNA primer, a DNA polymerase, normal deoxynucleosidetriphosphates (dNTPs), and modified nucleotides (dideoxyNTPs, ddNTP) that terminate DNA strand elongation.
- DNA fragments are labelled with a radioactive or fluorescent tag on the primer (1), in the new DNA strand with a labeled dNTP, or with a labeled ddNTP.

History of DNA sequencing technology

• First sequencing effort was taken on RNA. A first bacteria RNA sequence was reported by Walter Fiers at Univ of Ghent in Belgium in 1972.

• In 1970s, several notable advancements in DNA sequencing were made.

• In 1977, at MRC in UK, Frederick Sanger and colleagues developed a rapid DNA sequencing method, which becomes most widely-used sequencing method for approximately 25 years.

• The longer read (fragment) lengths in Sanger methods display significant advantages over other sequencing methods especially in terms of sequencing repetitive regions of the genome.

• Good for de novo sequencing and in sequencing highly rearranged genome segments, e.g. cancer genome, etc.

• In 2000s, “Next Generation” sequencing method invented, which allows sequencing in parallel and in large scale.
Automation of Sanger DNA sequencing

1. Reaction mixture
   - Primer and DNA template
   - DNA polymerase
   - ddNTPs with fluorochromes
   - dNTPs (dATP, dCTP, dGTP, and dTTP)

2. Primer elongation and chain termination

3. Capillary gel electrophoresis separation of DNA fragments

4. Laser detection of fluorochromes and computational sequence analysis

Ref: http://en.wikipedia.org/wiki/Microfluidic_Sanger_Sequencing
NGS Technology
How Does Next Generation Sequencing Work?

Schematic Overview of Large-scale of DNA Sequencing

Ref: http://www.broadinstitute.org/gatk/guide/topic?name=best-practices
Commercialized NGS
The gDNA (genomic DNA) is first fragmented into a library of small segments that can be uniformly and accurately sequenced in millions of parallel reactions. The newly identified strings of bases, called reads, are then reassembled using a known reference genome as a scaffold (resequencing), or in the absence of a reference genome (de novo sequencing). The full set of aligned reads reveals the entire sequence of each chromosome in the gDNA sample.

Next-Generation Sequencing Technology – Illumina Approach *in details*

1. **SONICATION**
   - Genomic DNA is fragmented into 100-500 base pair fragments by sonication to create a library.

2. **FRAGMENT END REPAIR**
   - Sonication creates frayed DNA ends which must be blunted or repaired.

3. **A-TAILING AND ADAPTER LIGATION**
   - Adapters are ligated to each end of the A-tailed DNA fragment.

4. **QC CHECK**
   - The electropherogram shows the size and concentration of the final library. This library size also confirms the ligation of adapters.

Ref: U.S. Department of Energy Office of Science
The Illumina approach

5. **cBOT CLUSTER GENERATION SYSTEM**
   - Adapter
   - DNA
   - Adapter
   - Dense lawn of adapters

Sodium hydroxide creates single-stranded DNA. Randomly bind these single-stranded DNA to the top and bottom of each channel in the flow cell.

6. **BRIDGE FORMATION**
   - Free DNA end binds to complimentary primer to form a bridge.

7. **BRIDGE AMPLIFICATION**
   - Add unlabeled nucleotides and enzyme to initiate solid-phase bridge amplification. Fragments become double-stranded DNA bridges. Thirty-five (35) cycles of amplification create clusters of identical DNA fragments.

8. **FINISHED FLOWCELL**
   - By completion of amplification, several million dense clusters of single-stranded DNA have been generated in each channel of the flow cell with a sequencing primer attached.

Ref: U.S. Department of Energy Office of Science
The Illumina approach

9 DNA SEQUENCING
To initiate the first sequencing cycle and determine the first base, all four labeled reversible terminators and DNA polymerase enzyme are first added. Only one base can incorporate at a time.

11 SEQUENCING-BY-SYNTHESIS
In the first cycle, the first base is incorporated. Its identity is determined by the signal given off and then recorded. In subsequent cycles, the process of adding sequencing reagents, removing unincorporated bases and capturing the signal of the next base to identify is repeated.

10 BASE CALLING
Lasers excite the fluorescent tags and the images are captured via CCD camera. The identity of the first base in each cluster is recorded, and then the fluorescent tag is removed.

12 DUAL FLOW CELLS
Once the top surface of the flow cell channel has been scanned, the imaging step is repeated on the bottom surface.

Ref: U.S. Department of Energy Office of Science
Whole genome shotgun sequencing & Hierarchical shotgun sequencing

Whole genome shotgun sequencing
- The entire genome is sheared randomly into small fragments
- Sequencing fragments → reads
- Reassembling reads

Hierarchical shotgun sequencing
- The genome is first broken into larger segments.
- After the order of these segments is deduced, they are further sheared into fragments.
- Sequencing fragments → reads
- Reassembling reads.

Ref: http://en.wikipedia.org/wiki/Shotgun_sequencing
Enormous pile of short reads from NGS

Mapping and alignment algorithms

Reference genome

Reads mapped to reference

Strand | Sequence
--- | ---
Original | AGCATGCTGCAGTCATGCTTAGGCTA
First shotgun sequence | AGCATGCTGCAGTCATGCT-\cdots-\cdots-TAGGCTA
Second shotgun sequence | AGCATG-\cdots-\cdots-CTGAGTCATGCTTAGGCTA
Reconstruction | AGCATGCTGCAGTCATGCTTAGGCTA

Non-reference bases are colored; reference bases are grey

Clean C/T heterozygote

Depth of coverage

First and second read from the same fragment

Individual reads aligned to the genome

Reference genome

Ref: http://www.broadinstitute.org/gatk/guide/topic?name=best-practices
A survey of sequence alignment algorithms for next-generation sequencing

Heng Li and Nils Homer
Submitted: 3rd March 2010; Received (in revised form): 14th April 2010

Abstract
Rapidly evolving sequencing technologies produce data on an unparalleled scale. A central challenge to the analysis of this data is sequence alignment, whereby sequence reads must be compared to a reference. A wide variety of alignment algorithms and software have been subsequently developed over the past two years. In this article, we will systematically review the current development of these algorithms and introduce their practical applications on different types of experimental data. We come to the conclusion that short-read alignment is no longer the bottleneck of data analyses. We also consider future development of alignment algorithms with respect to emerging long sequence reads and the prospect of cloud computing.

In a couple of years, however, long reads will dominate again and programs developed for short reads will not be applicable; long-read alignment and de novo assembly will become crucial. This approach, but establishing a cloud computing framework requires the efforts of the entire community. Furthermore, data transfer bottlenecks and leased storage have yet to be proved cost-effective for cloud computing.

Another trend of development is the simultaneous alignment against multiple genomes. Li et al. [69] have found the presence of extensive novel sequences absent from the human reference genome, which may lead to the loss of information when reads are aligned to a single genome. In the light

Key Points
- The advent of new sequencing technologies paves the way for various biological studies, most of which involves sequence alignment in an unparalleled scale.
- The development of alignment algorithms has been successful and short-read alignment against a single reference is not the bottleneck in data analyses any more.
- With the increasing read lengths produced by the new sequencing technologies, we expect further development in multi-reference alignment, long-read alignment and de novo assembly.
Next Generation Sequencing

High pass

- ~30x reads
- Excellent sensitivity for hetero- and homozygotes
- High depth allows excellent genotype calling

Low pass

- ~4x reads
- Heterozygotes can be mistaken for homozygotes due to sampling
- Variants missed by sampling
- Significantly better power to detect homozygous sites

Exom Capture

- 150x reads
- Little off-target coverage

Ref: http://www.broadinstitute.org/gatk/guide/topic?name=best-practices
• A sequencing run can be tailored to produce more or less data, zoom in with high resolution on particular regions of the genome, or provide a more expansive view with lower resolution.

• The term **coverage** generally refers to the **average number of sequencing reads that align to each base within the sample DNA**. For example, a whole genome sequenced at 30× coverage means that, on average, each base in the genome was covered by 30 sequencing reads.

• In cancer research, **somatic mutations** may only exist within a small proportion of cells in a given tissue sample. Using **mixed-cell samples**, the region of DNA harboring the mutation must be sequenced at very high levels of coverage, upwards of **1000×**, to detect these low frequency mutations within the cell population.

• For **genome-wide variant discovery**, a researcher would likely choose a much lower coverage level to sequence at lower resolution, but process larger sample numbers to achieve greater statistical power within a given population of interest.

**Data Analysis Algorithms**
The algorithms for sequencing data analysis differ depending on a given application.

For **de novo** sequencing, it requires specialized assembly of sequencing reads. For RNA-Seq, it requires to quantify read counts to provide information about gene expression levels.
An analogy of NGS strategy
Next Generation Sequencing

Base calling (vendor tool)

FASTQ file: raw NGS reads

Alignment or assembly

SAM/BAM file: aligned NGS reads

Variant calling

VCF file: genomic variation

.sam file: Uncompressed text file

.bam file: Compressed and indexed file

VCF Files can be block-compressed and indexed.

Ref: http://www.broadinstitute.org/gatk/guide/topic?name=best-practices
NGS Data Processing
Workflow using GATK

Ref: http://www.broadinstitute.org/gatk/guide/

* RR Compression – compress file size through reducing reads (remove redundant info)
Workflow using GATK – mapping and dedupping

Ref:

Non-GATK
- Raw Reads
  - Map To Reference
  - Mark Duplicates
  - Indel Realignment
  - Base Recalibration
  - RR Compression
  - Analysis-Ready Reads

GATK
- A-R Reads
  - Variant Calling
    - Raw Variants
      - SNPs
      - Indels
  - Variant Recalibration
    - Genotype Refinement
    - Variant Evaluation
  - Analysis-Ready Variants
    - SNPs
    - Indels
Workflow using GATK

Reference genome

Enormous pile of short reads from NGS

→ Map reads to reference with **BWA**
All later steps assume that reads are placed in the right location and represent that region of the genome.

→ Mark duplicates with **Picard tools**
Duplicates originate mostly from DNA prep methods and cause biases that skew variant calling results.

→ **mapping quality (MQ)**
Mapping algorithms account for this by choosing the most likely placement

Workflow using GATK

Typical workflow using BWA to map paired-end data

1. Align separately
   - FWD reads
     - fwd.fq
     - bwa aln
       - ref.fasta
       - fwd.fq
       - > fwd.sai
     - Index of FWD read positions
       - fwd.sai
   - REV reads
     - rev.fq
     - bwa aln
       - ref.fasta
       - rev.fq
       - > rev.sai
     - Index of REV read positions
       - rev.sai

2. Combine all
   - bwa sampe
     - ref.fasta
     - fwd.sai
     - rev.sai
     - fwd.fq
     - rev.fq
     - > mydata.sam
   - All reads aligned to reference
     - mydata.sam

Ref: http://www.broadinstitute.org/gatk/guide/
**FASTQ: raw unaligned reads from sequencer**

Each block has 4 elements (in 4 lines):
- Sequence Name (read name, group, etc.)
- Sequence
- + (optional: Sequence name again)
- Associated quality score.

**Example**
@SEQ_ID
GATTTGGGGTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!""*((((****++)(%%++)(%%%%).1***-++!'**))**55CCF>>>>>>CCCCCCC65

Line 4 encodes the quality values for the sequence in Line 2, and must contain the same number of symbols as letters in the sequence.

```
@HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1
TTAATTGGTAAATAAATCTCCTAATAGTATGTTATTTGTTGAGGAGACTTTTGTGATGGCTTGAT
+HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1
efcffffffcfeefffcffffffdfff`feed`\_Ba^__[YBBBBBBBBBBRTT\]]}[ddd`ddd^dddadd^BBBBBBBBBBBBBBBBBBBBBBBB
```

**ASCII Table**

<table>
<thead>
<tr>
<th>Char</th>
<th>Dec</th>
<th>Char</th>
<th>Dec</th>
<th>Char</th>
<th>Dec</th>
<th>Char</th>
<th>Dec</th>
<th>Char</th>
<th>Dec</th>
<th>Char</th>
<th>Dec</th>
<th>Char</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>(sp)</td>
<td>32</td>
<td>(</td>
<td>40</td>
<td>0</td>
<td>48</td>
<td>8</td>
<td>56</td>
<td>@</td>
<td>64</td>
<td>H</td>
<td>72</td>
<td>P</td>
<td>80</td>
</tr>
<tr>
<td>!</td>
<td>33</td>
<td>)</td>
<td>41</td>
<td>1</td>
<td>49</td>
<td>9</td>
<td>57</td>
<td>A</td>
<td>65</td>
<td>I</td>
<td>73</td>
<td>Q</td>
<td>81</td>
</tr>
<tr>
<td>&quot;</td>
<td>34</td>
<td>*</td>
<td>42</td>
<td>2</td>
<td>50</td>
<td>:</td>
<td>58</td>
<td>B</td>
<td>66</td>
<td>J</td>
<td>74</td>
<td>R</td>
<td>82</td>
</tr>
<tr>
<td>#</td>
<td>35</td>
<td>+</td>
<td>43</td>
<td>3</td>
<td>51</td>
<td>;</td>
<td>59</td>
<td>C</td>
<td>67</td>
<td>K</td>
<td>75</td>
<td>S</td>
<td>83</td>
</tr>
<tr>
<td>$</td>
<td>36</td>
<td>,</td>
<td>44</td>
<td>4</td>
<td>52</td>
<td>&lt;</td>
<td>60</td>
<td>D</td>
<td>68</td>
<td>L</td>
<td>76</td>
<td>T</td>
<td>84</td>
</tr>
<tr>
<td>%</td>
<td>37</td>
<td>-</td>
<td>45</td>
<td>5</td>
<td>53</td>
<td>=</td>
<td>61</td>
<td>E</td>
<td>69</td>
<td>M</td>
<td>77</td>
<td>U</td>
<td>85</td>
</tr>
<tr>
<td>&amp;</td>
<td>38</td>
<td>.</td>
<td>46</td>
<td>6</td>
<td>54</td>
<td>&gt;</td>
<td>62</td>
<td>F</td>
<td>70</td>
<td>N</td>
<td>78</td>
<td>V</td>
<td>86</td>
</tr>
<tr>
<td>'</td>
<td>39</td>
<td>/</td>
<td>47</td>
<td>7</td>
<td>55</td>
<td>?</td>
<td>63</td>
<td>G</td>
<td>71</td>
<td>O</td>
<td>79</td>
<td>W</td>
<td>87</td>
</tr>
</tbody>
</table>

### Phred Quality Score

| Phred Quality Score 
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
</tr>
<tr>
<td>Probability of incorrect base call</td>
</tr>
<tr>
<td>Base call accuracy</td>
</tr>
</tbody>
</table>

An example of a base that has been given a very high Phred score of 50, indicating that there is 99.999% probability that this base has been correctly assigned.

An example of a base that has been given a Phred score of 10, indicating that there is only a 90% probability that this base has been correctly assigned.

An example of a base for which no Phred score could be calculated, since the sequencer could not determine which base was present (therefore, an ‘N’ was designated in the sequence).

---

Figure 1. An example of a DNA sequence tracing and the Phred score (grey bars) corresponding to each colored peak. The colored peaks on the trace correspond to each DNA letter. For example, ‘T’ bases are represented in red, and this sequence has four ‘T’ bases on a row, as viewed by the four red peaks in the sequence. The aqua horizontal line placed across the grey bars represents a Phred score of 20 which is considered an acceptable level of accuracy.

As indicated in Table 1, a Phred score of 20 corresponds to a 99% accuracy in the base call. Therefore, bars above this line indicate base calls that have a higher than 99% probability of being correct. Those below have less than a 99% probability of being correct. Sequence tracing program is courtesy of FinchTV (www.geospiza.com).
**Phred base-calling** is a computer program for identifying a base (nucleobase) sequence from a fluorescence "trace" data generated by an automated DNA sequencer that uses electrophoresis and 4-fluorescent dye method.

**Phred** was originally conceived in the early 1990s by Phil Green, then a professor at Washington University in St. Louis. "Phred reads DNA sequencer trace data, calls bases, assigns quality values to the bases, and writes the base calls and quality values to output files.” – from Phred (version 0.020425.c) documentation by Phil Green

**Phred** uses a four-phase procedure as outlined by Ewing *et al.* to determine a sequence of base calls from the processed DNA sequence tracing:

1. Predicted peak locations are determined, based on the assumption that fragments are relatively evenly spaced, on average, in most regions of the gel, to determine the correct number of bases and their idealized evenly spaced locations in regions where the peaks are not well resolved, noisy, or displaced (as in compressions)
2. Observed peaks are identified in the trace
3. Observed peaks are matched to the predicted peak locations, omitting some peaks and splitting others; as each observed peak comes from a specific array and is thus associated with 1 of the 4 bases (A, G, T, or C), the ordered list of matched observed peaks determines a base sequence for the trace.
4. The unmatched observed peaks are checked for any peak that appears to represent a base but could not be assigned to a predicted peak in the third phase and if found, the corresponding base is inserted into the read sequence.

The entire procedure is rapid, usually taking less than half a second per trace.

Workflow using GATK: mapping and duplicate marking

Typical workflow using Picard tools to mark duplicates et al.

Original SAM → sort → Ordered BAM

“dedup” → Dedupped BAM

Wham, BAM, thank you Picard!

Ref: http://www.broadinstitute.org/gatk/guide/
Workflow using GATK – realignment

Ref:
Workflow using GATK: indel-based realignment

- InDels in reads (especially near the ends) can trick the mappers into mis-aligning with mismatches.
- These artifactual mismatches can harm base quality recalibration and variant detection (unless a sophisticated caller like the Haplotype Caller is used).

Realignment around indels helps improve the accuracy of several of the downstream processing steps.

```
java --jar GenomeAnalysisTK.jar --T RealignerTargetCreator \
    -R human.fasta \
    -I original.bam \
    -known indels.vcf \
    -o realigner.intervals
```

```
java --jar GenomeAnalysisTK.jar --T IndelRealigner \
    -R human.fasta \
    -I original.bam \
    -known indels.vcf \
    -targetIntervals realigner.intervals \
    -o realigned.bam
```

Ref: http://www.broadinstitute.org/gatk/guide/
Workflow using GATK: indel-based realignment

Ref: http://www.broadinstitute.org/gatk/guide/

Several consecutive "SNPs" only found on reads ending on the right of the homopolymer.

Several consecutive "SNPs" only found on reads ending on the left of the homopolymer.

7bp "T" homopolymer run.
Local realignment uncovers the hidden indel in these reads and eliminates all the potential FP SNPs

Ref: http://www.broadinstitute.org/gatk/guide/
Local realignment identifies most parsimonious alignment along all reads at a problematic locus

1. Find the best alternate consensus sequence that, together with the reference, best fits the reads in a pile (maximum of 1 indel)

2. The score for an alternate consensus is the total sum of the quality scores of mismatching bases

3. If the score of the best alternate consensus is sufficiently better than the original alignments (using a LOD score), then we accept the proposed realignment of the reads

Ref: http://www.broadinstitute.org/gatk/guide/
Workflow using GATK: indel-based realignment

Before

After

1,000 Genomes Pilot 2 data, raw MAQ alignments

1,000 Genomes Pilot 2 data, after MSA

HiSeq data, raw BWA alignments

HiSeq data, after MSA

Ref: http://www.broadinstitute.org/gatk/guide/
Workflow using GATK – base recalibration

Non-GATK:
1. Raw Reads
2. Map To Reference
3. Mark Duplicates
4. Indel Realignment
5. Base Recalibration
6. RR Compression
7. Analysis-Ready Reads

GATK:
1. A-R Reads
2. Variant Calling
3. Raw Variants
   - SNPs
   - Indels
4. Variant Recalibration
5. Genotype Refinement
6. Variant Evaluation
7. Analysis-Ready Variants
   - SNPs
   - Indels

Ref:
Workflow using GATK: Base Quality Score Recalibration

- Quality scores are critical for all downstream analysis
- Systematic biases are a major contributor to bad calls
- The quality scores issued by sequencers are inaccurate and biased

![Original vs Recalibrated Quality Scores](original-vs-recalibrated.png)

- Analyze covariation among several features of a base, e.g.:
  - Reported quality score
  - Position within the read (machine cycle)
  - Preceding and current nucleotide (sequencing chemistry effect)
- Apply covariates through a piecewise tabular correction to recalibrate the quality scores of all reads in a BAM file.

Ref: http://www.broadinstitute.org/gatk/guide/
Workflow using GATK: Base Quality Score Recalibration

Base Quality Score Recalibration provides a calibrated error model from which to make mutation calls

Ref: http://www.broadinstitute.org/gatk/guide/
Workflow using GATK: Base Quality Score Recalibration

Base Recalibration steps / tools

1. **BaseRecalibrator (1)**
   - PrintReads
   - Processed data
   - `java -jar GenomeAnalysisTK.jar -T PrintReads
     -R human.fasta
     -I realigned.bam
     -B QSR recal.table
     -O recal.bam`

2. **BaseRecalibrator (2)**
   - AnalyzeCovariates
   - Output file
   - `java -jar GenomeAnalysisTK.jar -T BaseRecalibrator
     -R human.fasta
     -I realigned.bam
     -knownSites dbsnp137.vcf
     -knownSites gold.standard.indels.vcf
     -O recal.table`

3. **AnalyzeCovariates**
   - Plots
   - `java -jar GenomeAnalysisTK.jar -T BaseRecalibrator
     -R human.fasta
     -I realigned.bam
     -knownSites dbsnp137.vcf
     -knownSites gold.standard.indels.vcf
     -B QSR recal.table
     -O after recal.table`

4. **Analysis of Before/After plots**
   - Evaluate recalibration results
   - `java -jar GenomeAnalysisTK.jar -T AnalyzeCovariates
     -R human.fasta
     -before recal.table
     -after after recal.table
     -plots recal_plots.pdf`

Workflow using GATK – Reduce Reads (compression)

1. Raw Reads
2. Map To Reference
3. Mark Duplicates
4. Indel Realignment
5. Base Recalibration
6. RR Compression
7. Analysis-Ready Reads
8. A-R Reads
9. Variant Calling
10. Raw Variants
11. SNPs
12. Indels
13. Variant Recalibration
14. Genotype Refinement
15. Variant Evaluation
16. Analysis-Ready Variants
17. SNPs & Indels

Ref:
• The size of the BAM file is a major roadblock for data analysis scalability
• Huge size of a BAM file makes file transfer impractical, simple analysis time-consuming, and complex analysis non-viable.
• Data compression is to throw out redundant information

Ref: http://www.broadinstitute.org/gatk/guide/
Workflow using GATK – Calling Variants

Ref:
• Genetic variant or random machine noise?
  ➔ large scale Bayesian inference problem

There are two approaches:

1. Initial approach: very fast, independent base assumption
2. Evolved approach: more computationally intensive, involves local de-novo assembly of the variable region

Variant calling tools:

- **UnifiedGenotyper**
  Call SNPs and indels separately by considering each variant locus independently
  - Accepts any ploidy
  - Pooled calling
  - High sample numbers

- **HaplotypeCaller**
  Call SNPs, indels, and some SVs simultaneously by performing a local de-novo assembly
  - More accurate, especially for indels
  - Will eventually replace UG

Ref: http://www.broadinstitute.org/gatk/guide/
SNP and Indel calling is a large-scale Bayesian modeling problem

\[
Pr\{G|D\} = \frac{Pr\{G\} Pr\{D|G\}}{\sum_i Pr\{G_i\} Pr\{D|G_i\}}, \quad \text{[Bayes’ rule]}
\]

\[
Pr\{D|G\} = \prod_j \left( \frac{Pr\{D_j|H_1\}}{2} + \frac{Pr\{D_j|H_2\}}{2} \right)
\]

where \( G = H_1H_2 \)

- Inference: what is the genotype \( G \) of each sample given read data \( D \) for each sample?
- Calculate via Bayes’ rule the probability of each possible \( G \)
- Product expansion assumes reads are independent
- Relies on a likelihood function to estimate probability of sample data given proposed haplotype

Ref: http://www.broadinstitute.org/gatk/guide/
SNP genotype likelihoods

\[
\Pr\{D_j|H\} = \Pr\{D_j|b\}, \text{ [single base pileup]}
\]

\[
\Pr\{D_j|b\} = \begin{cases} 
1 - \epsilon_j & D_j = b, \\
\epsilon_j & \text{otherwise}.
\end{cases}
\]

- All diploid genotypes (AA, AC, ..., GT, TT) considered at each base
- Likelihood of genotype computed using only pileup of bases and associated quality scores at given locus
- Only “good bases” are included: those satisfying minimum base quality, mapping read quality, pair mapping quality, NQS

Ref: http://www.broadinstitute.org/gatk/guide/
Indel genotype likelihoods

\[ \Pr\{D_j|H\} = \sum_{\text{alignments } \pi \text{ of } D_j \text{ to } H} \Pr\{D_j, \pi\} \]

- Haplotypes \( H_i \) are discovered from indels in the reads
- Diploid genotypes \( G \) for all haplotype \( H_i \) combinations
- For each haplotype \( H_i \), calculate likelihood of each read \( D_j \) marginalizing over all possible alignments \( \pi \)
- Sum computed by a standard HMM with context-dependent affine gap penalties using haplotype and read bases and quality scores

Ref: http://www.broadinstitute.org/gatk/guide/
Multi-sample calling integrates per sample likelihoods to jointly estimate allele frequency of variation.

- Simultaneous estimation of:
  - Allele frequency (AF) spectrum $\Pr\{AF = i \mid D\}$
  - The probability that a variant exists $\Pr\{AF > 0 \mid D\}$
  - Assignment of genotypes to each sample

UnifiedGenotyper

- In discovery mode (default), outputs variants called above confidence thresholds

```java
java -jar GenomeAnalysisTK.jar -T UnifiedGenotyper \
-R human.fasta \
-I input.bam \
-o output.vcf \
-stand_call_conf 30 \
-stand_emit_conf 10
```

- Genotype Given Alleles (GGA) mode allows genotyping specific sites of interest

Ref: http://www.broadinstitute.org/gatk/guide/
Multi-sample calling integrates per sample likelihoods to jointly estimate allele frequency of variation.

- Simultaneous estimation of:
  - Allele frequency (AF) spectrum $\Pr\{AF = i \mid D\}$
  - The probability that a variant exists $\Pr\{AF > 0 \mid D\}$
  - Assignment of genotypes to each sample

A key step in NGS applications (e.g., RNA-Seq) is to map short reads to correct genomic locations within the source genome.

- most mappable short reads (>75%) based on available methods (e.g., Bowtie) align to a single genomic location with relatively high precision.
- a significant number of reads are mapped to more than one genomic location with similar fidelity, and these reads are called multireads.
- multireads disproportionally come from the genes with similar sequences (e.g., duplicated genes) and essentially determine their expression levels.

Ref: Biometrics, December 2011,
Let's say we have this example data:

```plaintext
sample1_lane1.fq
sample1_lane2.fq
sample2_lane1.fq
sample2_lane2.fq
```

1. Run all core steps per-lane once

Assuming one FASTQ file per lane of sequence data, just run each file through each pre-processing step individually: **map & dedup -> realign -> recal**. The example data becomes:

```plaintext
sample1_lane1.dedup.realn.recal.bam
sample1_lane2.dedup.realn.recal.bam
sample2_lane1.dedup.realn.recal.bam
sample2_lane2.dedup.realn.recal.bam
```

2. Merge lanes per sample

Once you have pre-processed each lane individually, you merge lanes belonging to the same sample into a single BAM file. The example data becomes:

```plaintext
sample1.merged.bam
sample2.merged.bam
```

3. Per-sample refinement

You can increase the quality of your results by performing an extra round of dedupping and realignment, this time at the sample level. It is not absolutely required and will increase your computational costs, so it's up to you to decide whether you want to do it on your data, but that's how we do it internally at Broad.

Sequencing as a covering problem
- DNA sequencing rely on reading small fragments of DNA and subsequently reconstructing these data to infer the original DNA target, either via assembly or alignment to a reference.
- How many small fragments of DNA are needed to cover the entire target region of a genome?

Elementary probability theory
- The probability of “covering” any given location on the target with one particular fragment is $L/G$, where $L=$ fragment length and $G=$ target length.
- The probability $P$ that the location is covered by at least one of $N$ fragments is

$$P = 1 - \left[ 1 - \frac{L}{G} \right]^N$$

when $N>>1$, it can be approximately expressed as

$$\left[ 1 - \frac{L}{G} \right]^N \sim \exp(-NL/G)$$

where $R=NL/G$ is called the redundancy, or “coverage”.

Lander-Waterman theory
Eric Lander and Michael Waterman in 1988 published a paper examining the covering problem from the standpoint of gaps. The main goal of a sequencing project is to close all gaps.

$$E(\#contigs) = Ne^{-R}$$

**Contig:** a succession of overlapping reads,
**Gaps:** regions not covered by any read

VCF $\rightarrow$ PLINK format using PSEQ
... focus ... [on] large-scale resequencing and genotyping projects ... independent of (but designed to be complementary to) ... **PLINK** package.

... to perform a range of basic, large-scale analyses ... purely on analysis of genotyping/phenotype data ...
Workflow using PSEQ – creating PLINK files from VCF files

1. Create a PSEQ project
   • `pseq proj_name new-project`

2. Import VCF file into the project
   • `pseq proj_name load-vcf --vcf data/*.vcf`

3. Convert to PLINK
   • `pseq proj_name write-ped --name mydata`
   Output: `mydata.tped, mydata.tfam`

4. PLINK processing
   • `plink --tfile mydata`

Glossary
• **Phred quality scores** were originally developed by the program **Phred** to help in the automation of **DNA sequencing** in the **Human Genome Project**. **Phred quality scores** are assigned to each **nucleotide** base call in automated sequencer traces. Phred quality scores have become widely accepted to characterize the quality of DNA sequences, and can be used to compare the efficacy of different sequencing methods. The Phred quality scores are widely used in standard Sanger format adopted in Illumina’s pipeline CASAVA.

• **Phred** first calculates several parameters related to **peak shape** and **peak resolution** at each base.

• **Phred** then uses these parameters to look up a corresponding **quality score in huge lookup tables**.

• These lookup tables were generated from sequence traces where the correct sequence was known, and are hard coded in Phred; different lookup tables are used for **different sequencing chemistries and machines**.

• An evaluation of the accuracy of **Phred quality scores** for a number of variations in sequencing chemistry and instrumentation showed that Phred quality scores are highly accurate

\[
Q = -10 \log_{10} P
\]

or

\[
P = 10^{-\frac{Q}{10}}
\]

• Sanger format can encode a **Phred quality score** from **0 to 93 using ASCII 33 to 126** (although in raw read data the Phred quality score rarely exceeds 60, higher scores are possible in assemblies or read maps). Also used in SAM format. Coming to the end of February 2011, Illumina's newest version (1.8) of their pipeline CASAVA will directly produce fastq in Sanger format, according to the announcement on seqanswers.com forum.

Ref: [http://en.wikipedia.org/wiki/Phred_quality_score](http://en.wikipedia.org/wiki/Phred_quality_score)
**FASTQ format** is a text-based format for storing both a biological sequence (usually nucleotide sequence) and its corresponding quality scores. Both the sequence letter and quality score are encoded with a single ASCII character for brevity. It was originally developed at the Wellcome Trust Sanger Institute to bundle a FASTA sequence and its quality data, but has recently become the *de facto* standard for storing the output of high throughput sequencing instruments such as the Illumina Genome Analyzer.

Example
```plaintext
@SEQ_ID
GATTTGGGGTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%++)(%%%)1***-++''))**55CCF>>>>>>CCCCCCCC65
```

Line 1 begins with a '@' character and is followed by a sequence identifier and an *optional* description (like a FASTA title line).
Line 2 is the raw sequence letters.
Line 3 begins with a '+' character and is *optionally* followed by the same sequence identifier (and any description) again.
Line 4 encodes the quality values for the sequence in Line 2, and must contain the same number of symbols as letters in the sequence.

E.g.
```plaintext
@HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1
TTAATTTGGAATTAATCTCCTAATAGCTTAGATNTAATCTTNNNNNTAGTTTTCTTGGATTTGTGATTTGAAAAAACATTTTTGTGATTTGCTTGTAT
+HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1
efcffccccfccccffccccffccccfddfd`feed`\_Ba_\^\_\[\{YBBBBBBBBBBBBB\}\]\}\]\}\]\}\]\}\]\}\]\}\]\}\]\}\]\}\]\}\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\}\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\}\}\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\]\}\]\]\]\]\}\]\]\]\]\}\]\}\]\}\]\]\}\]\]\]\]\]\]\]\}\]\}\]\]\}\]\]\]\}\]\}\]\}\]\]\]\]\}\]\}\]\}\]\}\]\]\}\]\}\}\]\}\}\]\}\}\}\]\}\}\]\}\}\]\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\]\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\]\}\]\}\]\]\}\]\]\}\]\]\}\]\]\}\]\]\]\}\]\]\]\]\]\]\}\]\]\}\]\]\]\}\]\}\]\}\]\}\}\]\}\}\]\}\}\}\}\]\}\]\}\}\}\}\}\}\}\}\}\}\}\]\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\\
Phred base-calling is a computer program for identifying a base (nucleobase) sequence from a fluorescence "trace" data generated by an automated DNA sequencer that uses electrophoresis and 4-fluorescent dye method.

Phred was originally conceived in the early 1990s by Phil Green, then a professor at Washington University in St. Louis. “Phred reads DNA sequencer trace data, calls bases, assigns quality values to the bases, and writes the base calls and quality values to output files.” – from Phred (version 0.020425.c) documentation by Phil Green

Phred uses a four-phase procedure as outlined by Ewing et al. to determine a sequence of base calls from the processed DNA sequence tracing:

1. Predicted peak locations are determined, based on the assumption that fragments are relatively evenly spaced, on average, in most regions of the gel, to determine the correct number of bases and their idealized evenly spaced locations in regions where the peaks are not well resolved, noisy, or displaced (as in compressions)
2. Observed peaks are identified in the trace
3. Observed peaks are matched to the predicted peak locations, omitting some peaks and splitting others; as each observed peak comes from a specific array and is thus associated with 1 of the 4 bases (A, G, T, or C), the ordered list of matched observed peaks determines a base sequence for the trace.
4. The unmatched observed peaks are checked for any peak that appears to represent a base but could not be assigned to a predicted peak in the third phase and if found, the corresponding base is inserted into the read sequence.

The entire procedure is rapid, usually taking less than half a second per trace.

Ref: http://en.wikipedia.org/wiki/Phred_base_calling
A **contig** (from *contiguous*) is a set of overlapping DNA segments that together represent a consensus region of DNA. In **bottom-up sequencing** projects, a contig refers to overlapping sequence data (reads); in **top-down sequencing** projects, contig refers to the overlapping clones that form a physical map of the genome that is used to guide sequencing and assembly. Contigs can thus refer both to overlapping DNA sequence and to overlapping physical segments (fragments) contained in clones depending on the context.

**SAM/BAM files** (more to see [http://samtools.sourceforge.net/SAMv1.pdf](http://samtools.sourceforge.net/SAMv1.pdf))
- SAM stands for Sequence Alignment/Map format.
- BAM is the binary version of a SAM file.

**Manipulating Alignment Files with samtools**

**Conversion:**

```
    samtools view -h file.bam > file.sam
    samtools view -b -S file.sam > file.bam
```

**Sorting:**

```
    samtools sort -m 1000000000 file.bam outputPrefix
```

where `–m` is the option for how much memory to use in sorting.

**Creating a BAM index:**

```
    samtools index sorted.bam
```

**Using BEDtools to perform basic analysis tasks**

Creating a Genome Coverage BedGraph:

```
    genomeCoverageBed -ibam SRR065240.notx.bam -bg -trackline -
    trackopts 'name="notx" color=250,0,0' > notx.bedGraph
```

*Note: BedGraph is a file format that allows display of continuous-valued data in a genome browser track.*