Introduction to second-generation sequencing
Review: DNA sequencing

• Technologies to determine the nucleotide sequences of a DNA molecule.

• Motivation: decipher the genetic codes hidden in DNA sequences for different biological processes.

• **Genome projects**: determine DNA sequences for different species, e.g., human genome project.

• **Genomic research** (in a nutshell): study the functions of DNA sequences and related components.
Sequencing technologies

• Traditional technology: **Sanger sequencing**.
  – Slow (low throughput) and expensive: it took Human Genome Project (HGP) 13 years and $3 billion to sequence the entire human genome.
  – Relatively accurate.

• New technology: different types of **high-throughput sequencing**.
Second generation sequencing

• Aka: high-throughput sequencing, next generation sequencing (NGS).

• Able to sequence large amount of short sequence segments in a short period:
  – high throughput: billions of sequences in a run.
  – Cheap: sequence entire human genome costs below one thousand dollars now.
  – short read length: up to several hundred bps.
## HiSeq X Instrument Performance Parameters

<table>
<thead>
<tr>
<th></th>
<th>Dual Flow Cell</th>
<th>Single Flow Cell</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Output per Run</strong></td>
<td>1.6-1.8 Tb</td>
<td>800-900 Gb</td>
</tr>
<tr>
<td><strong>Reads Passing Filter</strong></td>
<td>5.3-6 billion</td>
<td>2.6-3 billion</td>
</tr>
<tr>
<td><strong>Supported Read Length</strong></td>
<td></td>
<td>2 × 150 bp</td>
</tr>
<tr>
<td><strong>Run Time</strong></td>
<td></td>
<td>&lt; 3 days</td>
</tr>
<tr>
<td><strong>Quality Scores</strong></td>
<td></td>
<td>≥ 75% of bases above Q30 at 2 × 150 bp</td>
</tr>
<tr>
<td><strong>Supported Library Preparation</strong></td>
<td></td>
<td><strong>TruSeq DNA PCR-Free Library Prep Kit, TruSeq Nano DNA Library Prep Kit</strong></td>
</tr>
</tbody>
</table>
Available platforms

• Major player:
  – Illumina: HiSeq, MiSeq.
  – LifeTech: SOLiD, IonTorrent.
  – Roche 454.

• Others:
  – Pacific Bioscience (SMRT)
  – Oxford Nanopore
Second-generation sequencing technologies
Second-generation sequencing technologies

• Complicated and involves a lot of biochemical reactions.
  – Sequencing by synthesis.
  – Sequencing by ligation.
  – Pyrosequencing.

• In a nutshell:
  – Cut the long DNA into smaller segments (several hundreds to several thousand bases).
  – Sequence each segment: start from one end and sequence along the chain, base by base.
  – The process stops after a while because the noise level is too high.
  – Results from sequencing are many sequence pieces. The lengths vary, usually a few thousands from Sanger, and several hundreds from NGS.
  – The sequence pieces are called “reads” for NGS data.
Technology: Illumina/Solexa

1. Prepare genomic DNA
2. Attach DNA to surface
3. Bridge amplification
4. Fragment become double stranded
5. Denature the double stranded molecules
6. Complete amplification
7. Determine first base
8. Image first base
9. Determine second base
10. Image second base
11. Sequence reads over multiple cycles
12. Align data.

>50 million clusters/flow cell, each 1000 copies of the same template, 1 billion bases per run, 1% of the cost of capillary-based method.

Figure source: http://www.illumina.com/downloads/SS_DNAsequencing.pdf
ABI/SOLiD system

• Technology: sequencing by ligation.

• Unique 2-base encoding system: every dinucleotide is turned into a color.
SOLiD technology
Primer and ligation rounds

| Read Position | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 |
|---------------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1             |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Universal seq primer (n) 3' |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 2             |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Universal seq primer (n-1) 3' |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 3             |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Universal seq primer (n-2) 3' |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 4             |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Universal seq primer (n-3) 3' |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 5             |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Universal seq primer (n-4) 3' |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

*Indicates positions of interrogation*  

Ligation Cycle: 2, 3, 4, 5, 6, 7
2-base encoding

Possible Dinucleotides Encoded By Each Color

Template Sequence

Double Interrogation

With 2 base encoding each base is defined twice
SNP calling

- A SNP will cause two adjacent color changes.
- Not all color changes are valid.
Single-end vs. paired-end sequencing

- Sequence one or both ends of the DNA segments.
- **Single-end** sequencing: sequence one end of the DNA segment.
- **Paired-end** sequencing: sequence both ends of a DNA segments.
  - Result reads are “paired”, separated by certain length (the length of the DNA segments, usually a few hundred bps).
  - Paired-end data can be used as single-end, but contain extra information which is useful in some cases, e.g., detecting structural variations in the genome.
  - Modeling technique is more complicated.
Applications of Second-generation sequencing
Applications

• NGS has a wide range of applications.
  – DNA-seq: sequence genomic DNA.
  – RNA-seq: sequence RNA products.
  – ChIP-seq: detect protein-DNA interaction sites.
  – Bisulfite sequencing (BS-seq): measure DNA methylation strengths.
  – A lot of others.

• Basically replaced microarrays with better data: greater dynamic range and higher signal-to-noise ratios.
<table>
<thead>
<tr>
<th>Technology</th>
<th>Brief description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChIP-seq</td>
<td>Locate protein-DNA interaction or histone modification sites.</td>
</tr>
<tr>
<td>CLIP-seq</td>
<td>Map protein-RNA binding sites</td>
</tr>
<tr>
<td>RNA-seq</td>
<td>Quantify expression</td>
</tr>
<tr>
<td>SAGE-seq</td>
<td>Quantify expression</td>
</tr>
<tr>
<td>RIP-seq</td>
<td>Capture TF-bound transcripts</td>
</tr>
<tr>
<td>GRO-seq</td>
<td>Evaluate promoter-proximal pausing</td>
</tr>
<tr>
<td>BS-seq</td>
<td>Profile DNA methylation patterns</td>
</tr>
<tr>
<td>MeDIP-seq</td>
<td>Profile DNA methylation patterns</td>
</tr>
<tr>
<td>TAB-seq</td>
<td>Profile DNA hydroxyl-methylation patterns</td>
</tr>
<tr>
<td>MIRA-seq</td>
<td>Profile DNA methylation patterns</td>
</tr>
<tr>
<td>ChiRP-seq</td>
<td>Map IncRNA occupancy</td>
</tr>
<tr>
<td>DNase-seq</td>
<td>Identify regulatory regions</td>
</tr>
<tr>
<td>FAIRE-seq</td>
<td>Identify regulatory regions</td>
</tr>
<tr>
<td>FRT-seq</td>
<td>Quantify expression</td>
</tr>
<tr>
<td>Repli-seq</td>
<td>Assess DNA replication timing</td>
</tr>
<tr>
<td>MNase-seq</td>
<td>Identify nucleosome position</td>
</tr>
<tr>
<td>Hi-C</td>
<td>Infer 3D genome organization</td>
</tr>
<tr>
<td>ChIA-PET</td>
<td>Detect long distance chromosome interactions</td>
</tr>
<tr>
<td>4C-seq</td>
<td>Detect long distance chromosome interaction</td>
</tr>
<tr>
<td>Sono-seq</td>
<td>Map open-chromatin sites</td>
</tr>
<tr>
<td>NET-seq</td>
<td>Determine in vivo position of all active RNAP complexes.</td>
</tr>
<tr>
<td>NA-seq</td>
<td>Map Nuclease-Accessible Sites</td>
</tr>
</tbody>
</table>
DNA-seq

• Sequence the untreated genomic DNA.
  – Obtain DNA from cells, cut into small pieces then sequence the segments.

• Goals:
  – Compare with the reference genome and look for genetic variants:
    • Single nucleotide polymorphisms (SNPs)
    • Insertions/deletions (indels),
    • Copy number variations (CNVs)
    • Other structural variations (gene fusion, etc.).
  – De novo assembly of a new genome.
Variations of DNA-seq

• Targeted sequencing, e.g., exome sequencing.
  – Sequence the genomic DNA at targeted genomic regions.
  – Cheaper than whole genome DNA-seq, so that money can be spent to get bigger sample size (more individuals).
  – The targeted genomic regions need to be “captured” first using technologies like microarrays.

• Metagenomic sequencing.
  – Sequence the DNA of a mixture of species, mostly microbes, in order to understand the microbial environments.
  – The goal is to determine number of species, their genome and proportions in the population.
  – De novo assembly is required. But the number and proportions of species are unknown, so it poses challenge to assembly.
RNA-seq

- Sequence the “transcriptome”: the set of RNA molecules.

- Goals:
  - Catalogue RNA products.
  - Determine transcriptional structures: alternative splicing, gene fusion, etc.
  - Quantify gene expression: the sequencing version of gene expression microarray.
ChIP-seq

- Chromatin-Immunoprecipitation (ChIP) followed by sequencing (seq): sequencing version of ChIP-chip.
- Used to detect locations of certain “events” on the genome:
  - Transcription factor binding.
  - DNA methylations and histone modifications.
- A type of “captured” sequencing. ChIP step is to capture genomic regions of interest.
Second-generation sequencing data analyses
Workflow of second generation sequencing data analysis

Raw images

Fluorescent intensities

sequence reads

aligned reads

contigs

alignement

de novo assembly

variant calling (DNA seq)

DE/splicing (RNA seq)

peak/DMR detection (ChIP/MeDIP- seq)

...
Imaging analysis

• Extract intensity values from images.
  – On Illumina and SOLiD systems, there are four images per cycle, one for a nucleotide/color.

• Similar to that in microarrays.

• Involves many statistical methods to extract signals from noisy data.

• Results of the imaging analysis: a 3-dimensional matrix: nreads x 4 x nbases.
Base calling

- For each read, at each position, convert four fluorescent intensities (continuous) into a base or color (categorical).
- It’s a classification problem.

<table>
<thead>
<tr>
<th>Base</th>
<th>Base1</th>
<th>Base2</th>
<th>Base3</th>
<th>Base4</th>
<th>Base5</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.290</td>
<td>0.046</td>
<td>0.014</td>
<td>0.026</td>
<td>0.010</td>
</tr>
<tr>
<td>C</td>
<td>0.014</td>
<td>0.654</td>
<td>0.132</td>
<td>0.803</td>
<td>0.006</td>
</tr>
<tr>
<td>G</td>
<td>0.062</td>
<td>0.009</td>
<td>0.001</td>
<td>0.016</td>
<td>0.712</td>
</tr>
<tr>
<td>T</td>
<td>0.016</td>
<td>0.010</td>
<td>0.455</td>
<td>0.046</td>
<td>0.768</td>
</tr>
</tbody>
</table>

\[ \downarrow \]

ACTCT...
Example of base calling method: Alta-Cyclic for Illumina data

The training process (green arrows) starts with creation of the training set, beginning with sequences generated by the standard Illumina pipeline, by linking intensity reads and a corresponding genome sequence (the 'correct' sequence). Then, two grid searches are used to optimize the parameters to call the bases. After optimization, a final SVM array is created, each of which corresponds to a cycle. In the base-calling stage (blue arrows), the intensity files of the desired library undergo deconvolution to correct for phasing noise using the optimized values and are sent for classification with the SVM array. The output is processed, and sequences and quality scores are reported.

Example of base calling method: RSOLiD for SOLiD data

Observation: Bases called are unbalanced toward the end of the reads.

Fluorescent intensity distributions

Cycle 5, manufacturer

Cycle 15, manufacturer

Cycle 25, manufacturer

Cycle 35, manufacturer

Density

-1 0 1

-1 0 1

-1 0 1

-1 0 1

FTC
Cy3
TXR
Cy5
Quantile normalization method

- Assume $f_{jc} = (1-p_c) f_{0j} + p_c f_{1j}$
  - $f_{jc}$: intensity distributions for color $c$ at the $j$th cycle.
  - $f_{0j}, f_{1j}$: intensity distributions for background and signal at the $j$th cycle, independent of colors.
  - $p_c$: proportion of dinucleotide corresponding to color $c$ in the sample, independent of cycle.

- To do:
  1. Estimate $f_{0j}, f_{1j}$ and $p_c$ and create target distributions.
  2. Quantile normalize intensities to targets.
Before and after normalization

Before normalization

After normalization

Color call proportions

cycles

cycles
## Quantile normalization improves the alignment results

<table>
<thead>
<tr>
<th></th>
<th>Before QN</th>
<th>After QN</th>
<th>% improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sample 1</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Mapped Reads</td>
<td>660850</td>
<td>710226</td>
<td>7.47</td>
</tr>
<tr>
<td>0 mismatches</td>
<td>246542</td>
<td>281590</td>
<td>14.22</td>
</tr>
<tr>
<td>total reads</td>
<td>169708</td>
<td>180460</td>
<td>6.34</td>
</tr>
<tr>
<td>1 mismatch</td>
<td>134467</td>
<td>138811</td>
<td>3.23</td>
</tr>
<tr>
<td>2 mismatches</td>
<td>110133</td>
<td>109365</td>
<td>-0.70</td>
</tr>
<tr>
<td><strong>Sample 2</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Mapped Reads</td>
<td>14090775</td>
<td>14985313</td>
<td>6.35</td>
</tr>
<tr>
<td>0 mismatches</td>
<td>5490005</td>
<td>6202116</td>
<td>12.97</td>
</tr>
<tr>
<td>total reads</td>
<td>3511552</td>
<td>3679413</td>
<td>4.78</td>
</tr>
<tr>
<td>1 mismatch</td>
<td>2794532</td>
<td>2829559</td>
<td>1.25</td>
</tr>
<tr>
<td>2 mismatches</td>
<td>2294686</td>
<td>2274225</td>
<td>-0.89</td>
</tr>
<tr>
<td>3 mismatches</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Raw sequence reads from second generation sequencing after base calling

- Large text file (millions of lines) with simple format.
  - Most frequently used: fasta/fa format for storing the sequences, or fastq format storing both the sequence and corresponding quality scores.
- fasta format:

```plaintext
>5_143_428_832
GATATTGTAGCATAACGCAACTTGGGAGGTGAGCTT
>5_143_984_487
GTTTTCATGCCTCCAAATCTTGGAGGCTTTTTATG
>5_143_963_690
GGTATATGCACAAAATGAGATGCTTGCTTATCAACA
>5_143_957_461
GGAGGGTGTCAATCCTGACGGTTATTTCCTAGACAA
>5_143_808_403
GATAACCGCATAAGCTCTTGGGAAGAGATTCTGTCT
```
fastq format

@HWI-EAS165:1:1:50:908:1
CTGCGGTCTCTAAAGTGCCATCTCATTTTGTATGCTAGTGCTGGA+
BCCBCB8ABBBBBBB:BC=8@BBA:@BB@BBBABB<9BBAC;A<CBABAB<#

@HWI-EAS165:1:1:50:0:1
NCAACCCCAACAGTAAATATGAAAAAATAAAAACACTAAAAACCAGGAGCTGAAGGG+
#BABBBBBBB@08<@?A@7:A@CCBCCCCBBBCCBB=?BBBB07@B=A>2
@HWI-EAS165:1:1:50:708:1
GGTCAGCATGCTTCTCTTAAGTGCTTGCAACAGCTAGGCTCTGCCTATGGG+
BB@A;B>0@@=BB=BBBB?A>@@B?ABBA=A?@@>@@A=??>?A=B@@AB
@HWI-EAS165:1:1:50:1494:1
CTGTTGTCACAAGCAGGCTCTCTGTGTGGACTCACCAGACACTGTCTATT+
BCBB@AB@1ABBBBBBBABB?BBBABB<A?AA>BB@?1ABBA@BBBA@;B>:
Sequence alignment and assembly

- **Sequence a known genome? --- Alignment**
  - Use the known genome (called “reference genome”) as a blue print.
  - Determine where each read is located in the reference genome.

- **Sequence a whole new genome? --- Assembly**
  - New genome: a species with unknown genome, or the genome is believed to be very different from reference (e.g., cancer).
  - Basically the short reads are “stitched” together to form long sequences called “contigs”.
  - Overlaps among sequence reads are required, so it needs a lot of reads (deep coverage).
  - More computationally intensive.
Alignment

• Need: sequence reads file and a reference genome.
• It is basically a string search problem: where is the short (50-letter) string located within the reference string of 3 billion letters.
• Brute-force searching is okay for a single read, but computationally infeasible to alignment millions of reads.
• Clever algorithms are needed to preprocess the reference genome (indexing), which is beyond the scope of this class.
Popular general alignment software

- Bowtie: fast, but less accurate.
- BWA (Burrows-Wheeler Aligner): same algorithm as bowtie, but allow gaps in alignments.
  - about 5-10 times slower than bowtie, but provide better results especially for paired end data.
- Maq (Mapping and Assembly with Qualities): with SNP calling capabilities.
- ELAND: Illumina’s commercial software.
Other technology-specific alignment software

• RNA-seq:
  – Tophat
  – STAR

• Bisulfite sequencing:
  – Bismark
  – BSMAP
  – Merman
Bowtie is an ultrafast, memory-efficient short read aligner. It aligns short DNA sequences (reads) to the human genome at a rate of over 25 million 35-bp reads per hour. Bowtie indexes the genome with a Burrows-Wheeler index to keep its memory footprint small: typically about 2.2 GB for the human genome (2.9 GB for paired-end).
Use bowtie: build alignment index

- Alignment index files are built based on reference genome (can be download as text files from UCSC).
- Note that pre-built indexes for many genomes are available from bowtie page, check that before building your own index.
- Command example for Human hg18 genome. Assume we have the hg18 sequence file ready called hg18.fa:
  
  ```bash
  bowtie-build hg18.fa hg18
  ```

- Results: several ebwt files.
- Tips: the index files can be stored in a common place and shared among colleagues.
Use bowtie: alignment

• Test whether it works:
  bowtie -c hg18 GGTATATGCAACAAATGAGATGCTTGCTTA

• Align a read files
  bowtie -v 3 -f hg18 reads.fa reads.map
bowtie: commonly used parameters

- **Input file format:**
  - `-q`: query input files are FASTQ .fq/.fastq (default)
  - `-f`: query input files are (multi-)FASTA .fa/.mfa
  - `-r`: query input files are raw one-sequence-per-line

- **Aligment:**
  - `-v`: allowing v mismatches.
  - `-5`: ignoring some based from 5’ end.
  - `-3`: ignoring some based from 3’ end.

- **Output format:**
  - `-S`: output in SAM (sequence alignment map) format.

- **Example:** input is a single fa file, allowing 3 mismatches, ignore 5 bases from 3’ end, output in SAM format:

```
bowtie -v 3 -3 5 -S hg18 reads.fa reads.sam
```
Output from bowtie

- **SAM format**

```
@HD     VN:1.0  SO:unsorted
@sreads.fa reads.sam
5_143_428_832   4       *       0       0       *       *       0       0       GATATTGTAGCATAACGCAACTTGGGAGGTGAGCTT
5_143_984_487   0       phage   3948    255     36M     *       0       0       GTTTTCATGCCTCCAAATCTTGGAGGCTTTTTATG
5_143_963_690   0       phage   3503    255     36M     *       0       0       GATATTGAAACTGAGTGTTACGTGAATCTGGAGAGCT
5_143_981_626   16      phage   1522    255     36M     *       0       0       TCCTCCTGAGACTGAGCTTTCTCGCCAAATGACGAC
5_143_470_717   16      phage   2061    255     36M     *       0       0       ATGCGCCTTCGTATGTTTCTCCTGCTTATCACCTTC
5_143_992_626   4       *       0       0       *       *       0       0       GCCCAAGGCGGCGTTAATGTTTGGAGAAAAG
5_143_400_771   0       phage   3816    255     36M     *       0       0       GATATTTTCATGGAATTGATAAAGCTGTTGCCGAT
```}

- **Bowtie format**

```
5_143_428_832   4       *       0       0       *       *       0       0       GATATTGTAGCATAACGCAACTTGGGAGGTGAGCTT
5_143_984_487   0       phage   3948    255     36M     *       0       0       GTTTTCATGCCTCCAAATCTTGGAGGCTTTTTATG
5_143_963_690   0       phage   3503    255     36M     *       0       0       GATATTGAAACTGAGTGTTACGTGAATCTGGAGAGCT
5_143_981_626   16      phage   1522    255     36M     *       0       0       TCCTCCTGAGACTGAGCTTTCTCGCCAAATGACGAC
5_143_470_717   16      phage   2061    255     36M     *       0       0       ATGCGCCTTCGTATGTTTCTCCTGCTTATCACCTTC
5_143_992_626   4       *       0       0       *       *       0       0       GCCCAAGGCGGCGTTAATGTTTGGAGAAAAG
5_143_400_771   0       phage   3816    255     36M     *       0       0       GATATTTTCATGGAATTGATAAAGCTGTTGCCGAT
```
Once the reads are aligned

- Downstream analyses depend on purpose.
  - We will cover the analyses for RNA-seq, ChIP-seq, and BS-seq in next several lectures.

- Often one wants to manipulating and visualizing the alignment results. There are several useful tools:
  - file manipulating (format conversion, counting, etc.): samtools/Rsamtools, BEDTools, bamtools, IGV tools.
  - Visualizing: samtools (text version), IGV (Java GUI).
samtools

- samtools provide various utilities for manipulating alignments in the SAM format, including sorting, merging, indexing and generating alignments in a per-position format.
- Command line driven, meaning one needs to type command in a terminal window.
  - Installation could be tricky. Needs to install extra tools on Windows or Mac, such as Cygwin and perl on Windows and Xcode on Mac.
- Main functionalities:
  - view: SAM<->BAM conversion
  - sort: sort alignment file
  - mpileup: multi-way pileup
  - depth: compute the coverage depth
  - tview: text alignment viewer
  - index: index alignment
samtools: generate sorted, indexed bam files

- **BAM file**: binary SAM. Smaller file sizes and faster operations.

- To convert from sam to bam:
  ```
  samtools view -bS reads.sam > reads.bam
  ```

- Sort and index bam file. This sorts the reads by chromosome and position and makes subsequence analysis easier.
  ```
  samtools sort reads.bam reads.sorted
  samtools index reads.sorted.bam
  ```
samtools: SNP calling

• SNP calling in samtools takes two steps:
  1. pileup the reads: all reads information are summarized at all base pair positions.
  2. Consensus variant calling using bcftools.

• Example:

samtools mpileup -uf ref.fa reads.sorted.bam>reads.pileup
bcftools view -v reads.pileup > SNP.vcf
Another useful software: BEDTools

- A set of commands to manipulate BED/GFF/VCF files.
- Conversion tools: `pairToBed(BAM)`, `bamToBed`, `bedToBam`, etc.
- Counting tools: `coverageBed(BAM)`, `windowBed(BAM)`
- Others: `sortBed`, `overlap`, etc.
Bioconductor package: Rsamtools

• Provide functions to import BAM files to R.
• There are many tools (samtools, BEDTools, bamtools) available to convert different formats (BED, SAM, fasta, fastq, etc.) to BAM.
• Read alignment results should always be saved in BAM format because they are smaller and faster.
Read in a BAM file

```r
> bamFile="reads.sorted.bam"
> bam <- scanBam(bamFile)
> names(bam[[1]])
[1] "qname"  "flag"   "rname"  "strand" "pos"    "qwidth" "mapq"   "cigar"
[9] "mrnm"   "mpos"   "isize"  "seq"    "qual"

This gives the available information in the BAM file. One can specify what to read in (to save time and memory):

```r
> what <- c("rname", "strand", "pos", "qwidth") ## fields to read in
> param <- ScanBamParam(what = what)
> bam <- scanBam(bamFile, param=param)[[1]]
> names(bam)
[1] "rname"  "strand" "pos"    "qwidth"
> bam$pos[1:10]
 [1] 1 2 3 3 4 4 4 4 4 5
> bam$strand[1:10]
[1] + + + + - - - - +
Levels: + - *
```
Summarize the read counts

• Remember each aligned read can be treated as a genomic interval. So the results from scanBam can be used to construct an GRanges object (of millions of intervals):

```r
> GRanges.reads = GRanges(seqnames = Rle(bam$rname),
                          ranges = IRRanges(bam$pos, width = bam$qwidth))
```

• Then it becomes very handy, for example, we can:
  – compute genome coverage:

```r
> cc = coverage(IRange.reads)
```
  – count number of reads in intervals (such as genes):

```r
> countOverlaps(genes, GRanges.reads)
```
An example: obtaining RNA-seq reads mapped to exons and introns

```r
library(GenomicRanges)
library(GenomicFeatures)
library(Rsamtools)

## get gene annotation, and extract exons/introns
refGene.hg18=makeTranscriptDbFromUCSC(genom="hg18", tablename="refGene")
ex=exonsBy(refGene.hg18, "tx")
intr=intronsByTranscript(refGene.hg18)

## read in RNA-seq BAM file
what=c("rname", "strand", "pos", "qwidth")
TSS.counts=NULL
param=ScanBamParam(what = what)
bam=scanBam("RNA-seq.bam", param=param)[[1]]
IRange.reads=GRanges(seqnames=Rle(bam$rname),
    ranges=IRanges(bam$pos, width=bam$qwidth))

## obtain counts
counts.exon=countOverlaps(ex, IRange.reads)
counts.intron=countOverlaps(intr, IRange.reads)
```
Visualization of sequencing data – Integrated Genome Viewer (IGV)

• “The **Integrative Genomics Viewer (IGV)** is a high-performance visualization tool for interactive exploration of large, integrated datasets. It supports a wide variety of data types including sequence alignments, microarrays, and genomic annotations.”

• Written in Java and runs on all OS.

• Very versatile and fast.

• Ability to connect to data server and display some public data (from ENCODE, broad, etc.)
Aligned reads on IGV
ChIP-seq data on IGV
RNA-seq junction reads on IGV
Review

• We’ve covered
  – basics of second generation sequencing technologies.
  – Some base calling methods.
  – Alignment using bowtie
  – Manipulation of alignment results using samtools
  – Import alignment into R using Rsamtools.
  – Visualization using IGV.