Introduction to NGS Analysis

Anna Battenhouse
abattenhouse@utexas.edu

February, 2019

Associate Research Scientist
Center for Systems and Synthetic Biology (CSSB)
   Ed Marcotte & Vishwanath Iyer labs

Center for Biomedical Research Support (CBRS)
   Bioinformatics Consulting Group (BCG)
   Biomedical Research Computing Facility (BRCF)
   Genome Sequencing & Analysis Facility (GSAF)
NGS Workflow

上游过程
- 实验设计
- DNA/RNA 提取
- 库制备
- 下一代测序
- 原始读取物交付

核心过程
- 原始读取 QC
- 有参考？
  - 是
    - 读取映射到参考
    - 对齐
      - 基本分析
        - (例如：覆盖度，基因)
  - 否
    - 原始读取交付

下游过程
- 对齐（基因组或转录本）
- 度量和 QC
- 下游过程
  - 差异分析
  - 标注
  - 动态分析
  - 自定义分析

 niềm: fastq
ılmış: BAM
var: bed, gff, vcf, etc.
1. Overview of sequencing technologies
2. NGS concepts and terminology
3. The FASTQ format and raw data QC & preparation
4. Alignment to a reference
Part 1: Overview of Sequencing Technologies

- High-throughput (“next gen”) sequencing
- Illumina short-read sequencing
- Long read sequencing
**NGS Workflow**

**upstream processes**

- Experimental design
- DNA/RNA isolation
- Library preparation
- Next-gen sequencing
- Delivery of raw reads

**core processes**

- QC raw read sequences
- Map reads to reference
- Alignment metrics & QC
- Basic analysis (e.g. coverage, genes)
- Further analysis & significance determination (e.g. FPKM, peak or variant calls)
- Confident calls

**has reference?**

- Yes
  - Reference assembly
  - Reference
  - Assembly (genome or transcriptome)

- No
  - QC raw read sequences
  - QC raw read sequences

**downstream processes**

- Metrics & QC
- Assembly
- Metrics & QC
- Metrics & QC
- Diff. analysis
- Annotation
- Motif analysis
- Custom analysis
“Next Generation” sequencing

- Massively parallel
  - simultaneously sequence “library” of millions of different DNA fragments

- **PCR colony clusters** generated
  - individual template DNA fragments titrated onto a flowcell to achieve inter-fragment separation
  - PCR “bridge amplification” creates clusters of identical molecules

- **Sequencing by synthesis**
  - fluorescently-labeled dNTs added
  - incorporation generates signal
  - flowcell image captured after each cycle
  - images computationally converted to base calls (including a quality score)

---

[https://www.nature.com/articles/nbt1486](https://www.nature.com/articles/nbt1486)
“Next Generation” sequencing (2nd generation)

- **Pro’s:**
  - much faster!
  - much lower cost!
  - both deeper and wider coverage!

- **Con’s:**
  - data deluge!
  - storage requirements!
  - analysis lags!

---

**Cost per Genome**

- Moore’s Law

**SRA database growth**

- $10^{16}$ bases (10 PetaBase) & growing

**SRA = Sequence Read Archive**

(NCBI public sequence database)
Illumina sequencing

1. Library preparation
2. *Cluster generation via bridge amplification*
3. Sequencing by synthesis
4. Image capture
5. Convert to base calls

---

**Note**

- 2 PCR amplifications performed
  1. during *library preparation*
  2. during *cluster generation*
- *amplification always introduces bias!*

---

Short Illumina video  
(https://tinyurl.com/hvnmwjb)

---

Illumina sequencing

1. Library preparation
2. Cluster generation via bridge amplification
3. Sequencing by synthesis
4. Image capture
5. Convert to base calls

http://www.cegat.de/
Multiplexing

- Illumina sequencers have one or more flowcell “lanes”, each of which can generate millions of reads
  - ~20M reads/lane for MiSeq, ~250M reads/lane for HiSeq
- When less than a full flowcell lane is needed, multiple samples with different barcodes (a.k.a. indexes) can be run on the same lane
  - 6-8 bp library barcode attached to DNA library fragments
  - data from sequencer must be demultiplexed to determine which reads belong to which library

https://doi.org/10.2147/BLCTT.S51503
Long read sequencing

- Short read technology limitations
  - PCR amplification bias
  - short reads are difficult to assemble
    - e.g., too short to span a long repeat region
  - difficult to detect large structural variations like inversions

- Newer “single molecule” sequencing
  - sequences single molecules, not clusters
  - allows for much longer reads (multi-Kb!)
    - no signal wash-out due to lack of synchronization among cluster molecules
  - but: reads have high error rate
    - 10-15% vs <1% for Illumina
  - one amplification usually still required (during library prep)
Long read sequencing

- Oxford Nanopore Ion Torrent system
  - [https://nanoporetech.com/](https://nanoporetech.com/)
  - DNA “spaghetti’s” through tiny protein pores
  - Addition of different bases produces different pH changes
    - measured as different changes in electrical conductivity
  - MinION is hand-held, starter kit costs ~$1,000 – including reagents!
Long read sequencing

- PacBio SMRT system

  - Sequencing by synthesis in *Zero-Mode Waveguide* (ZMW) wells
  - DNA is circularized then repeatedly sequenced to achieve “consensus”
Part 2: NGS Concepts & Terminology

- Experiment types & library complexity
- Sequencing terminology
- Sequence duplication issues
NGS Workflow

**core processes**

- QC raw read sequences
- map reads to reference
- alignment metrics & QC
- basic analysis (e.g., coverage, genes)
- further analysis & significance determination (e.g., FPKM, peak or variant calls)
- confident calls

**upstream processes**

- experimental design
- DNA/RNA isolation
- library preparation
- next-gen sequencing
- delivery of raw reads

**downstream processes**

- assembly (genome or transcriptome)
- metrics & QC
- differential analysis
- annotation
- motif analysis
- custom analysis

**has reference?**

- yes
  - reference assembly
  - fasta
  - bed, gff, vcf, etc.

- no
  - QC raw read sequences
Library complexity (diversity)
is a measure of the number of
distinct molecular species in the library.

Many different molecules $\rightarrow$ high complexity
Few different molecules $\rightarrow$ low complexity

The number of different molecules in a library depends on enrichment performed during library construction.
Popular Experiment Types

- **Whole Genome sequencing (WGS)**
  - *library*: all genomic DNA
  - *complexity*: high (fragments must cover the entire genome)
  - *applications*: genome assembly, variant analysis

- **Exome sequencing (WXS)**
  - *library*: DNA from eukaryotic exonic regions (uses special kits)
  - *complexity*: high/med (only ~5% of eukaryotic genome is in exons)
  - *applications*: polymorphism/SNP detection; genotyping

- **RNA-seq**
  - *library*: extracted RNA converted to cDNA
  - *complexity*: med/high (only a subset of genes are expressed in any given tissue)
  - *applications*: differential gene expression

- **Amplicon panels (targeted sequencing)**
  - *library*: DNA from a set of PCR-amplified regions using custom primers
  - *complexity*: very low (only 1 to a few thousand different library molecules)
  - *applications*: genetic screening panels; metagenomics; mutagenesis
Library complexity is primarily a function of experiment type

higher complexity

Less enrichment for specific sequences

Genomic

Bisulfite-seq

Exon capture

RNA-seq

ChIP-seq

Amplicons

lower complexity

Higher diversity of library molecules

Lower sequence duplication expected

More sequencing depth required

... as well as ...

- Genome size & sequencing depth
- Library construction skill & luck!

Lower diversity of library molecules

Higher sequence duplication expected

Less sequencing depth required

More enrichment for specific sequences
Read types

**single-end**

Independent reads

**paired-end**

two inwardly oriented reads separated by ~200 nt

**mate-paired**

two outwardly oriented reads separated by ~3000 nt
With paired-end sequencing, keep in mind the distinction between

- the library *fragment* from your library that was sequenced
  - also called *inserts*
- the *sequence reads* (R1s & R2s) you receive
  - also called *tags*
- an R1 and its associated R2 form a *read pair*
  - a readout of part (or all) of the fragment molecule
Single end vs Paired end

- **single end** (SE) reads are less expensive

- **paired end** (PE) reads can be mapped more reliably
  - especially against lower complexity genomic regions
    - an unmapped read can be “rescued” if its mate maps well
  - they provide more bases around a locus
    - e.g. for analysis of polymorphisms
  - actual fragment sizes can be easily determined
    - from the alignment records for each dual-mapping “proper pair”
  - also help distinguish the true complexity of a library
    - by clarifying which *fragments* are duplicates (vs *read* duplicates)

- **but** PE reads are more expensive – and larger
  - more storage space and processing time required

- General guidelines
  - use PE for high location accuracy and/or base-level sensitivity
  - use SE for lower-complexity, higher duplication experiments
Sequence Duplication

- The set of sequences you receive can contain exact duplicates

- Duplication can arise from:
  1. sequencing of species enriched in your library (*biological – good*)
     - each read comes from a different DNA molecule (cluster)
  2. sequencing of artifacts (*technical – bad*)
     - differentially amplified PCR species (PCR duplicates)
       - recall that 2 PCR amplifications are performed w/Illumina sequencing
     - *cannot tell which using “standard” sequencing methods!*

- Standard best practice is to “mark duplicates” during initial processing
  - then decide what to do with them later…
    - e.g. retain (use all), remove (use only non-duplicates), dose (use some)

- Different experiment types have different *expected* duplication
Expected sequence duplication is primarily a function of experiment type

Less enrichment for specific sequences
- genomic
- bisulfite-seq
- exon capture
- RNA-seq
- ChIP-seq
- amplicons

Higher complexity

Higher diversity of library molecules
Lower sequence duplication expected
More sequencing depth required

… as well as…
- genome size & sequencing depth
- library construction skill & luck!

More enrichment for specific sequences
- genomic
- bisulfite-seq
- exon capture
- RNA-seq
- ChIP-seq
- amplicons

Lower complexity

Lower diversity of library molecules
Higher sequence duplication expected
Less sequencing depth required
Read vs Fragment duplication

- Consider the 4 fragments below
  - 4 R1 reads (pink), 4 R2 reads (blue)

- Duplication when only 1 end considered
  - A1, B1, C1 have identical sequences, D1 different
    - 2 unique + 2 duplicates = 50% duplication rate
  - B2, C2, D2 have identical sequences, A2 different
    - 2 unique + 2 duplicates = 50% duplication rate

- Duplication when both ends considered
  - fragments B and C are duplicates (same external sequences)
    - 3 unique + 1 duplicate = 25% duplication rate

\[\text{A1} \quad \text{B1} \quad \text{C1} \quad \text{D1} \quad \text{A2} \quad \text{B2} \quad \text{C2} \quad \text{D2}\]
Molecular Barcoding

- Resolves ambiguity between biological and technical (PCR amplification) duplicates
  - adds secondary, *internal* barcodes to *pre-PCR* molecules
  - combination of barcodes + insert sequence can provide accurate quantification
  - but requires specialized pre- and post-processing

<table>
<thead>
<tr>
<th>actual count</th>
<th>Original library</th>
<th>naive count</th>
<th>Amplified library</th>
<th>barcode-aware count</th>
</tr>
</thead>
<tbody>
<tr>
<td>A : 2</td>
<td>molecule A-1</td>
<td>A : 3</td>
<td>molecule A-1</td>
<td>A : 2</td>
</tr>
<tr>
<td>B : 1</td>
<td>molecule B</td>
<td>B : 3</td>
<td>molecule B</td>
<td>B : 1</td>
</tr>
</tbody>
</table>

A/B 2/1
Part 3: The FASTQ format, Data QC & preparation

- FASTA and FASTQ formats
- QC of raw sequences with FastQC tool
- Dealing with adapters
NGS Workflow

**core processes**

- QC raw read sequences
  - has reference?
    - yes
    - no
  - map reads to reference
    - reference assembly
      - assembly
        - (genome or transcriptome)
      - metrics & QC
        - fasta
        - alignment metrics & QC
        - basic analysis
          - (e.g. coverage, genes)
          - downstream processes
            - differential analysis
            - annotation
            - motif analysis
            - custom analysis
        - further analysis & significance determination
          - (e.g. FPKM, peak or variant calls)
        - confident calls
          - delivery of raw reads
          - next-gen sequencing
            - library preparation
              - DNA/RNA isolation
                - experimental design
                  - upstream processes
                    - experimental design
                      - DNA/RNA isolation
                        - library preparation
                          - next-gen sequencing
                            - delivery of raw reads
FASTQ format

- Text format for storing sequence and quality data

- 4 lines per sequence:
  1. *read name*
  2. *called base sequence* (ACGTN)
     - always 5’ to 3’; *usually* excludes 5’ adapter
  3. *optional read name*
  4. *base quality scores encoded as text characters*

- FASTQ representation of a single, 50 base R1 sequence

```plaintext
@HWI-ST1097:97:D0WW0ACXX:8:1101:2007:2085 1:N:0:ACTTGAATTCTCCAAGATTTGGCAATGATGAGTACAAATTATATGCCCCAATTTACA+
?@@?DD;?;FF?HHBB+:ABECGHDHDF4?FGIGACFDFH;FHEIIIB9?
```
FASTQ read names

- Illumina Fastq read names encode information about the source cluster
  - unique identifier ("fragment name") begins with @, then:
    - sequencing machine name
    - lane number
    - flowcell coordinates
  - a space separates the name from extra read information:
    - end number (1 for R1, 2 for R2)
    - two quality fields (N = not QC failed)
    - library barcode sequence
- R1, R2 reads have the same fragment name
  - this is how the reads are linked to model the original fragment molecule

@HWI-ST1097:97:D0WW0ACXX:8:1101:2007:2085 1:N:0:ACTTGA
@HWI-ST1097:97:D0WW0ACXX:8:1101:2007:2085 2:N:0:ACTTGA
FASTQ quality scores

- Base qualities expressed as **Phred** scores
  - log scaled, *higher = better*
  - $20 = 1/10^2 = 1/100$ errors, $30 = 1/10^3 = 1/1000$ errors

  \[
  \text{Probability of Error} = 10^{-Q/10}
  \]

- Integer Phred score converted to Ascii character (add 33)

  [Table of quality characters and ASCII values]

<table>
<thead>
<tr>
<th>Quality character</th>
<th>ASCII Value</th>
<th>Base Quality (Q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>!&quot;#$%&amp;'()七星、—/.0123456789:;&lt;=?&gt;@ABCDEFGHIJKLMNOPQRSTUVWXYZ</td>
<td>33 43 53 63 73</td>
<td>0 10 20 30 40</td>
</tr>
</tbody>
</table>

  [ASCII table for quality characters]
Raw sequence quality control

- Critical step! Garbage in = Garbage out
  - general sequence quality
    - base quality distributions
    - sequence duplication rate
  - trim 3’ adapter sequences?
    - important for RNAseq
  - trim 3’ bases with poor quality?
    - important for de novo assembly
  - other contaminants?
    - biological – rRNA in RNAseq
    - technical – samples sequenced w/other barcodes
3’ Adapter contamination

A. reads short compared to fragment size (no contamination)

B. Reads long compared to library fragment (3’ adapter contamination)

The presence of the 3’ adapter sequence in the read can cause problems during alignment, because it does not match the genome.
FastQC

- Quality Assurance tool for FASTQ sequences
  http://www.bioinformatics.babraham.ac.uk

- Most useful reports:
  1. **Per-base sequence quality Report**
     - Should I trim low quality bases?
  2. **Sequence duplication levels Report**
     - How complex is my sequence library?
  3. **Overrepresented sequences Report**
     - Do I need to remove adapter sequences?
1. FastQC Per-base sequence quality report

Quality scores across all bases (Sanger / Illumina 1.9 encoding)
2. FastQC Sequence duplication report
Yeast ChIP-seq

For every 100 unique sequences there are:
~12 sequences w/2 copies
~1-2 with 3 copies

Ok – Some duplication expected due to IP enrichment
2. Sequence duplication report

Yeast ChIP-exo

For every 100 unique sequences there are:
~35 sequences w/2 copies
~22 with 10+ copies

Success! Protocol expected to have high duplication

Sequence Duplication Level >= 72.33%
Expected sequence duplication is primarily a function of experiment type

Less enrichment for specific sequences

- genomic
- bisulfite-seq
- exon capture
- RNA-seq
- ChIP-seq
- ChIP-exo
- amplicons

Higher diversity of library molecules
Lower sequence duplication expected
More sequencing depth required

More enrichment for specific sequences

Lower diversity of library molecules
Higher sequence duplication expected
Less sequencing depth required

… as well as…
- genome size & sequencing depth
- library construction skill & luck!
3. FastQC Overrepresented sequences report

- **FastQC** knows Illumina adapter sequences
- Here ~9-10% of sequences contain adapters
  - calls for adapter removal or trimming

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Count</th>
<th>Percentage</th>
<th>Possible Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGATCGGAAGAGCACACGTCTGAACCCAGCTCACCACACACAGATCTCGTGATG</td>
<td>60030</td>
<td>5.01369306977828</td>
<td>TruSeq Adapter, Index 1 (97% over 37bp)</td>
</tr>
<tr>
<td>GATCGGAAGAGCACACGTCTGAACCCAGCTCACCACACACAGATCTCGTGATG</td>
<td>42955</td>
<td>3.587592633884896</td>
<td>TruSeq Adapter, Index 1 (97% over 37bp)</td>
</tr>
<tr>
<td>CACACGTCTGAACCCAGCTCACCACACACAGATCTCGTGATGCGCGTCTTCTGCT</td>
<td>3574</td>
<td>0.2984997339894683</td>
<td>RNA PCR Primer, Index 40 (100% over 41bp)</td>
</tr>
<tr>
<td>CAGATCGGAAGAGCACACGTCTGAACCCAGCTCACCACACACAGATCTCGTGAT</td>
<td>2519</td>
<td>0.2103863540242356</td>
<td>TruSeq Adapter, Index 1 (97% over 37bp)</td>
</tr>
<tr>
<td>GAGATCGGAAGAGCACACGTCTGAACCCAGCTCACCACACACAGATCTCGTGAT</td>
<td>1251</td>
<td>0.1044832587543942</td>
<td>TruSeq Adapter, Index 1 (97% over 37bp)</td>
</tr>
</tbody>
</table>
3. Overrepresented sequences

- Here nearly 1/3 of sequences some type of non-adapter contamination
  - **BLAST** the sequence to identify it

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Count</th>
<th>Percentage</th>
<th>Possible Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAAGGCACGAGGAGACGAGCCGCTTTATCATCACGATAGGTTGCTAAGTG</td>
<td>5632816</td>
<td>32.03026785752871</td>
<td>No Hit</td>
</tr>
<tr>
<td>TATTCTGTGTCCTAGGCTAGAGAACAACACCCATCCATCCCAGAATTT</td>
<td>494014</td>
<td>2.8091456822607364</td>
<td>No Hit</td>
</tr>
<tr>
<td>TCAAAGGGAAGGCTTACGTTGAGATACCTTAGGACCCAGAGCAACAGGA</td>
<td>446641</td>
<td>2.539765344040083</td>
<td>No Hit</td>
</tr>
<tr>
<td>TAAAAGCAGCTCTGCGAAGGATATCTCGGTCTCGCATCGATGGAACACGT</td>
<td>179252</td>
<td>1.0192929387357474</td>
<td>No Hit</td>
</tr>
<tr>
<td>GAAGGTACAGGCGAGACGAGCCGTTTATCATCACGATAGGTTGCTAAGTG</td>
<td>171681</td>
<td>0.9762414422996221</td>
<td>No Hit</td>
</tr>
<tr>
<td>AACGACTCTCGGCAAGGATATCTCGGTCTCGCATCGATGGAACACGT</td>
<td>143415</td>
<td>0.8155105483274229</td>
<td>No Hit</td>
</tr>
<tr>
<td>AGAACATGAAACGTTAGCTCCAGACGATGGAGGAGCCTGGCTGCTG</td>
<td>111584</td>
<td>0.6345077504066322</td>
<td>No Hit</td>
</tr>
<tr>
<td>AAAACGACTCTCGGCAACGGATATCTCGGTCTCGCATCGATGGAACACGT</td>
<td>111255</td>
<td>0.6326369351474214</td>
<td>No Hit</td>
</tr>
<tr>
<td>ATTACGATAGGTGCAATGGAAGTGCGATGTATGCAGCTGAGGGA</td>
<td>73682</td>
<td>0.41898300890326096</td>
<td>No Hit</td>
</tr>
<tr>
<td>GAAGGTACACGCGAGACGAGCCGTATCATTACGATAGGTTGCTAAGGG</td>
<td>71661</td>
<td>0.4074908580252516</td>
<td>No Hit</td>
</tr>
<tr>
<td>GGATGCGATCTAAGGACGACTAATGCAACCGGATCCCATCGAAGAATCCGCA</td>
<td>69548</td>
<td>0.3954755612388914</td>
<td>No Hit</td>
</tr>
<tr>
<td>ATATTCTGTGTCCTAGGCGTACAGGAAACACCAATCCATCCGAACT</td>
<td>54017</td>
<td>0.30716057099328803</td>
<td>No Hit</td>
</tr>
</tbody>
</table>
Dealing with 3’ adapters

Three main options:

1. **Hard trim** all sequences by specific amount
   - e.g. trim 100 base reads to 50 bases
   - *Pro*: fast & easy to perform; trims low-quality 3’ bases
   - *Con*: removes information (bases) you might want

2. **Remove adapters** specifically
   - e.g. using specific tools
   - *Pro*: removes adapter contamination without losing sequenced bases
   - *Con*: requires knowledge of insert fragment structure & adapters

3. Perform a **local alignment** (vs **global**)
   - e.g. *bowtie2 --local* or *bwa mem*
   - *Pro*: mitigates adapter contamination while retaining full query sequence
   - *Con*: limited aligner support (but always needed for RNA-seq alignment)
FASTQ trimming

- **Tools:**

- **Features:**
  - hard-trim specific number of bases
  - trimming of low quality bases
  - specific trimming of adapters
  - support for trimming paired end read sets (except FASTX)
  - cutadapt has protocol for separating reads based on internal barcode
Local vs. Global alignment

- **Global** alignment
  - requires query sequence to map **fully** (end-to-end) to reference

- **Local** alignment
  - allows a **subset** of the query sequence to map to reference
    - “untemplated” adapter sequences will be “soft clipped” (ignored)

```
global (end-to-end) alignment of query
CACAAGTACAATTATACAC

local (subsequence) alignment of query
CTAGCTTATCGCCCTGAAGGACT
TACATA CACAAGTACAATTATACACAGACATTAGTTCTTATCGCCCTGAA AATTCTCC

reference sequence
```
Part 4: Alignment to a reference assembly

- Alignment overview & concepts
- Preparing a reference genome
- Alignment workflow steps
NGS Workflow

- **upstream processes**
  - experimental design
  - DNA/RNA isolation
  - library preparation
  - next-gen sequencing
  - delivery of raw reads

- **core processes**
  - QC raw read sequences
    - has reference?
      - yes: map reads to reference
      - no: reference assembly
        - fasta
          - assembly (genome or transcriptome)
            - metrics & QC
              - fasta
                - metrics & QC
                  - bed, gff, vcf, etc.
          - assembly
            - metrics & QC
              - bed, gff, vcf, etc.
                  - downstream processes
                      - differential analysis
                      - annotation
                      - motif analysis
                      - custom analysis

- **core processes**
  - map reads to reference
    - alignment metrics & QC
      - basic analysis
        - (e.g. coverage, genes)
          - further analysis & significance determination
            - (e.g. FPKM, peak or variant calls)
          - confident calls
Short Read Aligners

- Short read mappers determine placement of *query sequences* (your reads) against a known *reference*
  - **BLAST**:
    - one query sequence (or a few)
    - many matches for each
  - short read aligners
    - many millions of query sequences
    - want only one “best” mapping (or a few)

- Many aligners available! Two of the most popular
  - **bwa** (Burrows Wheeler Aligner) by Heng Li
  - **bowtie2** – part of the Johns Hopkins Tuxedo suite of tools
  - Given similar input parameters, they produce similar alignments
    - and both run relatively quickly
Mapping vs Alignment

- **Mapping** determines one or more *positions* (a.k.a. *seeds* or *hits*) where a read shares a short sequence with the reference.

- **Alignment** starts with the seed and determines how read bases are best *matched*, base-by-base, around the seed.

- Mapping quality and alignment scores are both reported:
  - High mapping quality ≠ High alignment score
  - **mapping quality** describes *positioning*
    - reflects the probability that the read is *incorrectly* mapped to the reported location
    - is a Phred score: \( P(\text{incorrectly mapped}) = 10^{-\text{mappingQuality}/10} \)
    - reflects the *complexity* or information content of the sequence (*mappability*)
  - **alignment score** describes *fit*
    - reflects the correspondence between the read and the reference sequence

Read 1

- Maps to one location *high mapping quality*
- Has 2 mismatches *low alignment score*

Read 2

- Maps to 2 locations *low mapping quality*
- Matches perfectly *high alignment score*

![Reference sequence](image-url)
Hash table enables lookup of exact matches.

<table>
<thead>
<tr>
<th>Subsequence</th>
<th>Reference Positions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATAGCTAATCCAAA</td>
<td>2341, 2617264</td>
</tr>
<tr>
<td>ATAGCTAATCCAAAT</td>
<td></td>
</tr>
<tr>
<td>ATAGCTAATCCAAAC</td>
<td>134, 13311, 732661,</td>
</tr>
<tr>
<td>ATAGCTATCCAAAG</td>
<td></td>
</tr>
<tr>
<td>ATAGCTAATCCATA</td>
<td></td>
</tr>
<tr>
<td>ATAGCTAATCCATT</td>
<td>3452</td>
</tr>
<tr>
<td>ATAGCTAATCCATC</td>
<td></td>
</tr>
<tr>
<td>ATAGCTATCCAAATG</td>
<td>234456673</td>
</tr>
</tbody>
</table>

Table is sorted and complete so you can jump immediately to matches. (But this can take a lot of memory.)

May include N bases, skip positions, etc.

Burrows-Wheeler transform compresses sequence.

Input
SIX.MIXED.PIXIES.SIFT.SIXTY.PIXIE.DUST.BOXES
Output
TEXYDST.E.IXIXIXXXSSMPPS.B...E.S.EUSFXDIIIOIIIT

Suffix tree enables fast lookup of subsequences.

Mapping via suffix array tree


Exact matches at all positions below a node.

Alignment via dynamic programming

- Dynamic programming algorithm (Smith-Waterman | Needleman-Wunsch)

![Dynamic Programming Alignment](image)

- **Alignment score** = \( \Sigma \)
  - match reward
  - base mismatch penalty
  - gap open penalty
  - gap extension penalty
- rewards and penalties may be adjusted for quality scores of bases involved

Reference sequence

<table>
<thead>
<tr>
<th>ATTTGC</th>
<th>GATCGGATGAAGACGAA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ATTTGC</th>
<th>GATCGGATGTTGACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ATTTGC</th>
<th>GATCGGATGTTGACTTTAA</th>
</tr>
</thead>
<tbody>
<tr>
<td>XX</td>
<td>XXii</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ATTTGC</th>
<th>GATCGGATGTTGACTTTAA</th>
</tr>
</thead>
<tbody>
<tr>
<td>XX</td>
<td>XXii</td>
</tr>
</tbody>
</table>
Paired End mapping

- Having paired-end reads improves mapping
  - mapping one read with high confidence anchors the pair
    - even when its mate read by itself maps several places equally

- Three possible outcomes of mapping an R1/R2 pair
  1. only one of a pair might map (*singleton/orphan*)
  2. both reads can map within the most likely distance range and with correct orientation (*proper pair*)
  3. both reads can map but with an unexpected insert size or orientation, or to different contigs (*discordant pair*)

- Insert size is reported in the alignment record
  - for both proper and discordant pairs
Alignment Workflow

1. Obtain reference genome
   - bwa index
   - bowtie2-build

2. Build aligner-specific reference index
   - custom binary index

3. QC & trim raw reads

4. Align reads to reference
   - bwa aln + bwa samse or sampe, bwa mem, or bowtie2

5. Convert SAM to BAM
   - samtools view

6. Sort BAM by position
   - samtools sort

7. Handle duplicates (optional)
   - Picard MarkDuplicates

8. Index BAM
   - samtools index

9. Alignment metrics & QC
   - samtools flagstat
   - samtools idxstat

Resources:
Obtaining/building a reference

- What is a reference?
  - *any set of named DNA sequences*
    - e.g. names are chromosome names
    - technically referred to as *contigs*
  - assembled genomes
    - Ensembl, UCSC, for eukaryotes
      - FASTA files (.fa, .fasta), + annotations (genome feature files, .gff)
    - NCBI RefSeq or GenBank for prokaryotes/microbes
  - any set of sequences of interest, e.g:
    - transcriptome (set of transcribed gene sequences)
    - rRNA genes (e.g. for filtering)

- Building a reference index
  - may take several hours to build
  - but you build each index once, use for multiple alignments
Alignment Workflow

1. Obtain reference genome
2. Build aligner-specific reference index
3. Align reads to reference
4. Convert SAM to BAM
5. Sort BAM by position
6. Handle duplicates (optional)
7. Index BAM
8. Alignment metrics & QC

**Tools:**
- BWA
- Bowtie2
- SAMtools
- Picard

**Related Resources:**
- [BWA Manual](http://bio-bwa.sourceforge.net/bwa.shtml)
**SAM / BAM file format**

- Aligners take FASTQ as input, output alignments in **Sequence Alignment Map (SAM)** format
  - plain-text file format that describes how reads align to a reference

- SAM and BAM are two forms of the same data
  - **BAM** – Binary Alignment Map
    - *same data* in a custom compressed (**gzip’ed**) format
    - *much* smaller than SAM files
    - when indexed, support fast random access (SAM files do not)

- SAM file consists of
  - a **header** (includes reference sequence names and lengths)
  - **alignment records**, one for each sequence read
    - alignments for R1 and R2 reads have *separate records*
    - records have 11 fixed fields + extensible-format **key:type:value** tuples
### SAM file format

#### Fixed fields (tab-separated)

<table>
<thead>
<tr>
<th>Col</th>
<th>Field</th>
<th>Type</th>
<th>Regexp/Range</th>
<th>Brief description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>QNAME</td>
<td>String</td>
<td>![?-A~]{1,255}</td>
<td>Query template NAME</td>
</tr>
<tr>
<td>2</td>
<td>FLAG</td>
<td>Int</td>
<td>[0,2^{16}-1]</td>
<td>bitwise FLAGs</td>
</tr>
<tr>
<td>3</td>
<td>RNAME</td>
<td>String</td>
<td>*</td>
<td>![()-+&lt;=&lt;&gt;-]{1,2}</td>
</tr>
<tr>
<td>4</td>
<td>POS</td>
<td>Int</td>
<td>[0,2^{29}-1]</td>
<td>1-based leftmost mapping POSition</td>
</tr>
<tr>
<td>5</td>
<td>MAPQ</td>
<td>Int</td>
<td>[0,2^{8}-1]</td>
<td>MAPping Quality</td>
</tr>
<tr>
<td>6</td>
<td>CIGAR</td>
<td>String</td>
<td>*</td>
<td>![0-9]+@[() ]</td>
</tr>
<tr>
<td>7</td>
<td>RNEXT</td>
<td>String</td>
<td>*=</td>
<td>![()-+&lt;=&lt;&gt;-]{1,2}</td>
</tr>
<tr>
<td>8</td>
<td>PNEXT</td>
<td>Int</td>
<td>[0,2^{29}-1]</td>
<td>Position of the mate/next segment</td>
</tr>
<tr>
<td>9</td>
<td>TLEN</td>
<td>Int</td>
<td>[-2^{29}+1,2^{29}-1]</td>
<td>observed Template LENgth</td>
</tr>
<tr>
<td>10</td>
<td>SEQ</td>
<td>String</td>
<td>*</td>
<td>![A-Za-z=.]</td>
</tr>
<tr>
<td>11</td>
<td>QUAL</td>
<td>String</td>
<td>![^-]</td>
<td>ASCII of Phred-scaled base QUALity+33</td>
</tr>
</tbody>
</table>

- **QNAME**: Query template NAME
- **FLAG**: bitwise FLAGs
- **RNAME**: Reference sequence NAME
- **POS**: 1-based leftmost mapping POSition
- **MAPQ**: MAPping Quality
- **CIGAR**: CIGAR string
- **RNEXT**: Ref. name of the mate/next segment
- **PNEXT**: Position of the mate/next segment
- **TLEN**: observed Template LENgth
- **SEQ**: segment SEQUENCE
- **QUAL**: ASCII of Phred-scaled base QUALity+33

#### Example Entries

**SRR030257.264529**

- **QNAME**: NC_012967
- **RNEXT**: 1521
- **PNEXT**: 29
- **TLEN**: 34M2S = 1564
- **SEQ**: CTGGCCATTATCTCGGTGGTATAGGACATGGCATGCCC
- **QUAL**: AAAAAA;AA;AAAAAA??A%.,?&'?3735',()*)
- **FLAG**: 99
- **RNEXT**: 29

**SRR030257.2669090**

- **QNAME**: NC_012967
- **RNEXT**: 1521
- **PNEXT**: 60
- **TLEN**: 36M = 1458
- **SEQ**: CTGGCCATTATCTCGGTGGTATAGGATGATATTGATGC
- **QUAL**: <<9<<<<AAAAA<<<<<<<<<<<<<<<<<<<<<<<<<<<<
- **FLAG**: 147
- **RNEXT**: 60

**Positive for plus strand reads**

**Negative for minus strand reads**
Alignment Workflow

- obtain reference genome
  - bwa index
  - bowtie2-build
- build aligner-specific reference index
  - custom binary index

QC & trim raw reads
- FastQC, cutadapt

align reads to reference
- bwa aln + bwa samse or sampe
  - bwa mem, or bowtie2

convert SAM to BAM
- samtools view

sort BAM by position
- samtools sort

handle duplicates (optional)
- Picard MarkDuplicates
- samtools rmdup

index BAM
- samtools index

alignment metrics & QC
- samtools flagstat
- samtools idxstat

http://broadinstitute.github.io/picard/
http://samtools.sourceforge.net/samtools.shtml
Sorting / indexing BAM files

- SAM created by aligner contains read records in name order
  - same order as read names in the input FASTQ file
    - R1, R2 have adjacent SAM records
    - SAM → BAM conversion does not change the name-sorted order

- Sorting BAM puts records in position (locus) order
  - by contig name then start position (leftmost)
  - sorting is very compute, I/O and memory intensive!
    - can take hours for large BAM

- Indexing a locus-sorted BAM allows fast random access
  - creates a small, binary alignment index file (.bai)
  - quite fast
Handling Duplicates

- Optional step, but very important for many protocols

Definition of alignment duplicates:
- single-end reads or singleton/discordant PE alignment reads
  - alignments have the same start positions
- properly paired reads
  - pairs have same external coordinates (5' + 3' coordinates of the insert)

Two choices for handling:
- **samtools rmdup** – *removes* duplicates entirely
  - faster, but data is lost
- **Picard MarkDuplicates** – *flags* duplicates only (0x400 bam flag)
  - slower, but all alignments are retained
- both tools are quirky in their own ways
Alignment Workflow

1. Obtain reference genome
   - bwa index
   - bowtie2-build

2. Build aligner-specific reference index
   - custom binary index

3. QC & trim raw reads
   - FastQC, cutadapt

4. Align reads to reference
   - bwa aln + bwa samse or sampe
   - bwa mem, or bowtie2

5. Convert SAM to BAM
   - samtools view

6. Sort BAM by position
   - samtools sort

7. Handle duplicates (optional)
   - Picard MarkDuplicates
   - samtools rmdup

8. Index BAM
   - samtools index

9. Alignment metrics & QC
   - samtools flagstat
   - samtools idxstat
Alignment metrics

- **samtools flagstat**
  - simple statistics based on alignment record flag values
  - total sequences (R1+R2), total mapped
  - number properly paired
  - number of duplicates (0 if duplicates were not marked)

161490318 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 secondary
0 + 0 supplementary
31602827 + 0 duplicates
158093331 + 0 mapped (97.90% : N/A)
161490318 + 0 paired in sequencing
80745159 + 0 read1
80745159 + 0 read2
153721151 + 0 properly paired (95.19% : N/A)
156184878 + 0 with itself and mate mapped
1908453 + 0 singletons (1.18% : N/A)
1061095 + 0 with mate mapped to a different chr
606632 + 0 with mate mapped to a different chr (mapQ>=5)
Alignment wrap up

- Many tools involved
  - choose one or two and learn their options well

- Many steps are involved in the full alignment workflow
  - important to go through manually a few times for learning
    - but gets tedious quickly!
  - best practice
    - automate series of complex steps by wrapping into a **pipeline script**
    - e.g. **bash** or **python** script
  - the Bioinformatics team has a set of pipeline scripts available at TACC
    - in shard project directory `/work/projects/BiolTeam/common/script/
    - **align_bowtie2_illumina.sh, align_bwa_illumina.sh, trim_adapters.sh**, etc.
Other NGS Resources at UT

- CCBB Summer School courses
  - 4 half-day sessions in May
  - Intro to NGS, RNAseq, several others
  - lots of hands-on, including w/TACC

- Genome Sequencing & Analysis Facility (GSAF)
  - Jessica Podnar, Director, gsaf@utgsaf.org

- Bioinformatics consultants
  - Dennis Wylie, Dhivya Arasappan, Benni Goetz, Anna

- Biomedical Research Support Facility (BRCF)
  - provides local compute and managed storage resources
  - https://wikis.utexas.edu/display/RCTFUsers

- BioITeam wiki – https://wikis.utexas.edu/display/bioiteam/