Introduction to NGS Analysis

Anna Battenhouse
abattenhouse@utexas.edu

March 8, 2022

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

Center for Systems and Synthetic Biology (CSSB)
Ed Marcotte & Vishwanath Iyer labs
Fastq
BAM
Bed, gff, vcf, etc.
QC raw read sequences

Has reference?

Yes
Map reads to reference
Reference assembly
Reference (genome or transcriptome)

No
Map reads to reference
Alignment metrics & QC
BAM
Bed, gff, vcf, etc.
Basic analysis (e.g. coverage, genes)
Further analysis & significance determination (e.g. FPKM, peak or variant calls)

Confident calls
Downstream processes

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

Upstream processes

Core processes

Assembly (genome or transcriptome)
Metrics & QC

Differential analysis
Annotation
Motif analysis
Custom analysis

NGS Workflow
Outline

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**

**Core Processes**
- QC raw read sequences
  - has reference?
    - yes
      - map reads to reference
        - reference assembly
          - fasta
        - alignment metrics & QC
          - BAM
        - basic analysis (e.g. coverage, genes)
          - bed, gff, vcf, etc.
      - no
        - downstream processes
          - assembly (genome or transcriptome)
          - metrics & QC
          - differential analysis
          - annotation
          - motif analysis
          - custom analysis
  - 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**
- experimental design
  - DNA/RNA isolation
    - library preparation
      - next-gen sequencing
        - delivery of raw reads

**Files & Formats**
- fastq
- reference (fasta)
- BAM
- bed, gff, vcf, etc.
- confident calls
- downstream processes
  - assembly (genome or transcriptome)
  - metrics & QC
  - differential 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)
  - results in 30 – 300 base “reads”
    - much shorter than Sanger sequencing

---

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!

**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, ~10G reads/lane for NovaSeq
- 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
Long read sequencing

- Short read technology limitations
  - 30 – 300 base reads (150 typical)
  - 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:
    - weaker signal leads to high error rate
      - 10% vs <1% for Illumina
    - and fewer reads are generated (~100 K)
Long read sequencing

- Oxford Nanopore ION technology systems (e.g. MinION)
  - [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!
    - inexpensive, but high error rates (~10%)
Long read sequencing

- PacBio SMRT system
  - Sequencing by synthesis in *Zero-Mode Waveguide* (ZMW) wells
  - DNA is circularized then repeatedly sequenced to achieve “consensus”
    - reduces error rate (~1-2%), but equipment *quite* expensive
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

**has reference?**

- yes
  - reference assembly
    - fasta

- no
  - assembly (genome or transcriptome)
  - metrics & QC
    - bed, gff, vcf, etc.

**upstream processes**

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

**downstream processes**

- differential analysis
- annotation
- motif analysis
- custom analysis
Library Complexity

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
<table>
<thead>
<tr>
<th>Type</th>
<th>Library construction</th>
<th>Applications</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whole genome (WGS)</td>
<td>• extract genomic DNA &amp; fragment</td>
<td>• Genome assembly</td>
<td>high</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Variant detection, genotyping</td>
<td></td>
</tr>
<tr>
<td>Bisulfite sequencing</td>
<td>• bisulfite treatment converts $C \rightarrow U$ but not $5\text{meC}$</td>
<td>• Methylation profiling (CpG)</td>
<td>high</td>
</tr>
<tr>
<td>RAD-seq, ddRAD</td>
<td>• restriction-enzyme digest DNA &amp; fragment</td>
<td>• Variant detection (SNPs)</td>
<td>high</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Population genetics, QTL mapping</td>
<td></td>
</tr>
<tr>
<td>Exome (WXS)</td>
<td>• capture DNA from exons only (manufacturer kits)</td>
<td>• Variant detection, genotyping</td>
<td>high-medium</td>
</tr>
<tr>
<td>ATAC-seq</td>
<td>• high-activity transposase cuts DNA &amp; ligates adapters</td>
<td>• Profile nucleosome-free regions (“open chromatin”)</td>
<td>medium-high</td>
</tr>
<tr>
<td>RNA-seq, Tag-seq</td>
<td>• extract RNA &amp; fragment</td>
<td>• Differential gene or isoform expression</td>
<td>medium, medium-low for Tag-seq</td>
</tr>
<tr>
<td></td>
<td>• convert to cDNA</td>
<td>• Transcriptome assembly</td>
<td></td>
</tr>
<tr>
<td>Transposon seq (Tn-seq)</td>
<td>• create library of transposon-mutated genomic DNA</td>
<td>• Characterize genotype/phenotype relationships w/high sensitivity</td>
<td>medium</td>
</tr>
<tr>
<td></td>
<td>• amplify mutants via Tn-PCR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ChIP-seq</td>
<td>• cross-link proteins to DNA</td>
<td>• Genome-wide binding profiles of transcription factors, epigenetic marks &amp; other proteins</td>
<td>medium (but variable)</td>
</tr>
<tr>
<td></td>
<td>• pull-down proteins of interest w/ specific antibody, reverse cross-links</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRO-seq</td>
<td>• isolate actively-transcribed RNA</td>
<td>• Characterize transcriptional dynamics</td>
<td>medium-low</td>
</tr>
<tr>
<td>RIP-seq</td>
<td>• like ChIP-seq, but with RNA</td>
<td>• Characterize protein-bound RNAs</td>
<td>low-medium</td>
</tr>
<tr>
<td>miRNA-seq</td>
<td>• isolate 15-25bp RNA band</td>
<td>• miRNA profiling</td>
<td>low</td>
</tr>
<tr>
<td>Amplicons</td>
<td>• amplify 1-1000+ genes/regions</td>
<td>• genotyping, metagenomics, mutagenesis</td>
<td>low</td>
</tr>
</tbody>
</table>
Library complexity is primarily a function of experiment type

- **Genomic**
  - Higher complexity
  - Higher diversity of library molecules
  - Lower sequence duplication expected
  - More sequencing depth required

- **ChIP-seq**
  - Lower complexity
  - Less enrichment for specific sequences
  - Lower diversity of library molecules
  - Higher sequence duplication expected
  - Less sequencing depth required

- **Exon capture**
  - More enrichment for specific sequences
  - ...as well as...
  - genome size & sequencing depth
  - library construction skill & luck!

- **RNA-seq**
  - More enrichment for specific sequences
  - Lower complexity
  - Higher diversity of library molecules
  - Lower sequence duplication expected

- **Bisulfite-seq**
  - Less enrichment for specific sequences
  - ...as well as...
  - genome size & sequencing depth
  - library construction skill & luck!

- **Amplicons**
  - More enrichment for specific sequences
  - Lower complexity
  - Lower diversity of library molecules
  - Higher sequence duplication expected
  - Less sequencing depth required
Read types

**single-end**
- R1
- independent reads

**paired-end**
- R1
- two inwardly oriented reads separated by ~200 nt
- R2

**mate-paired**
- alternative to long reads
- rarely used now (expensive)
- 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 experiment types
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
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.

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

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

More enrichment for specific sequences
- higher complexity
- lower complexity
Read vs Fragment duplication

- Consider the 4 “aligned” 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
Molecular Barcoding

- Resolves ambiguity between biological and technical (PCR amplification) duplicates
- adds secondary, internal barcodes to pre-PCR molecules
  - a.k.a UMIs (Unique Molecular Indexes)
- combination of barcodes + insert sequence provides accurate quantification
- but requires specialized pre- and post-processing
  - e.g. UMI de-duplication for 3’ Tag-seq, scRNA-seq

<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></td>
<td>molecule A-2</td>
<td></td>
<td>molecule A-2</td>
<td>B : 1</td>
</tr>
<tr>
<td>B : 1</td>
<td>molecule B</td>
<td>B : 3</td>
<td>molecule B</td>
<td>B : 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>molecule B</td>
<td>A/B 2/1</td>
</tr>
<tr>
<td>A/B</td>
<td>2/1</td>
<td>A/B 1/1</td>
<td></td>
<td>A/B 2/1</td>
</tr>
</tbody>
</table>
Single Cell sequencing

- Standard sequencing library starts with millions of cells
  - will be in different states unless synchronized
  - a heterogeneous “ensemble” with (possibly) high cell-to-cell variability

- **Single cell sequencing** technologies aim to capture this variability
  - examples:
    - cells in different layers/regions of somatic tissue
    - cells in different areas of a tumor
    - essentially a very sophisticated library preparation technique

- Typical protocol (RNA-seq)
  1. isolate a few thousand cells (varying methods)
  2. the single-cell platform partitions each cell into an emulsion droplet
    - e.g. 10x Genomics (https://www.10xgenomics.com/solutions/single-cell/)
  3. a different barcode is added to the RNA in each cell
  4. resulting library submitted for standard Illumina short-read sequencing
  5. custom downstream analysis links results to their cell (barcode) of origin
Part 3: The FASTQ format, Data QC & preparation

- The FASTQ format
- QC of raw sequences with FastQC tool
- Dealing with adapters
NGS Workflow

**core processes**

- fastq
- BAM
- bed, gff, vcf, etc.
- QC raw read sequences
  - has reference?
    - yes
      - map reads to reference
      - alignment metrics & QC
        - basic analysis
          - (e.g. coverage, genes)
  - no
    - reference assembly
      - fasta
        - metrics & QC
          - assembly (genome or transcriptome)

**upstream processes**

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

**downstream processes**

- delivery of raw reads
- experimental design
- DNA/RNA isolation
- library preparation
- next-gen sequencing
- metrics & QC
  - further analysis & significance determination
    - (e.g. FPKM, peak or variant calls)
  - confident calls
  - downstream processes
  - differential analysis
  - annotation
  - motif analysis
  - custom analysis

**status**

- has reference?
**FASTQ format**

- Text format for storing sequence and quality data
- 4 lines per sequence:
  1. **@read name** *(plus extra information after a space)*
     - R1 and R2 reads have the same read name
  2. **called base sequence** *(ACGTN)*
     - always 5’ to 3’; *usually* excludes 5’ adapter
  3. **+optional read information**
  4. **base quality scores encoded as text characters**

- FASTQ representation of a single, 50 base R2 sequence

```plaintext
@HWI-ST1097:97:D0WW0ACXX:4:1101:2007:2085 2:N:0:ACTTGA
ATTCTCCAAGATTTGGCAATGATGAGTACAATTATATGCCCCAATTTACA+
?@?DD;?;FF?HHBB+:ABECGHDHDFC4?FGIGACFDFFH;FHEIIIB9?
```
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

  Probability of Error $= 10^{-\frac{Q}{10}}$

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

<table>
<thead>
<tr>
<th>ASCII Value</th>
<th>Base Quality $(Q)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>33</td>
<td>0</td>
</tr>
<tr>
<td>43</td>
<td>10</td>
</tr>
<tr>
<td>53</td>
<td>20</td>
</tr>
<tr>
<td>63</td>
<td>30</td>
</tr>
<tr>
<td>73</td>
<td>40</td>
</tr>
</tbody>
</table>

http://www.asciitable.com/

?@@?DD;?;FF?HHBB+:ABECGHHDHDCF4?FGIGACFDFH;FHEIIIB9?
Raw sequence quality control

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

A. reads short compared to fragment size (no 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](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
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>AGATCGGAAGAGCAGTCGTCACTCAGTCACCTCAGAATCTCGTATG</td>
<td>60030</td>
<td>5.01369306977828</td>
<td>TruSeq Adapter, Index 1 (97% over 37bp)</td>
</tr>
<tr>
<td>GATCGGAAGAGCAGTCGTCACTCAGTCACCTCAGAATCTCGTATGC</td>
<td>42955</td>
<td>3.5875926338884896</td>
<td>TruSeq Adapter, Index 1 (97% over 37bp)</td>
</tr>
<tr>
<td>CACACGTCTGAACTCCAGTCACCTCAGAATCTCGTATGCCGTCCTCTGCT</td>
<td>3574</td>
<td>0.2984997339894683</td>
<td>RNA PCR Primer, Index 40 (100% over 41bp)</td>
</tr>
<tr>
<td>CAGATCGGAAGAGCAGTCGTCACTCAGTCACCTCAGAATCTCGTATG</td>
<td>2519</td>
<td>0.2103863542024236</td>
<td>TruSeq Adapter, Index 1 (97% over 37bp)</td>
</tr>
<tr>
<td>GAGATCGGAAGAGCAGTCGTCACTCAGTCACCTCAGAATCTCGTATG</td>
<td>1251</td>
<td>0.10448325887543942</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>GAAGGTCACGGCAGACGAGCCGTTTATCATTACGATAGGTTGCAAGTGGA</td>
<td>5632816</td>
<td>32.03026785752871</td>
<td>No Hit</td>
</tr>
<tr>
<td>TATTCTGGTGTCTCTAGCGTAGGGAACACACCAATCCATCCGAACCTT</td>
<td>494014</td>
<td>2.8091456822607364</td>
<td>No Hit</td>
</tr>
<tr>
<td>TCAAACGGAAAGGCTTACCGTGATACCTTGACCCAGACAGGAAGGA</td>
<td>446641</td>
<td>2.539765344040083</td>
<td>No Hit</td>
</tr>
<tr>
<td>TAAAACGACTCTCGGCAACGGATACTCGGCTCTCGCATCGATGAAGAAC</td>
<td>179252</td>
<td>1.0192929387357474</td>
<td>No Hit</td>
</tr>
<tr>
<td>GAAGGTCACGGCAGACGAGCCGTTTATCATTACGATAGGTTGCAAGTGGA</td>
<td>171681</td>
<td>0.9762414422996221</td>
<td>No Hit</td>
</tr>
<tr>
<td>AACGACCTCTCGGCAACGGATACTCGGCTCTCGCATCGATGAAGAACGA</td>
<td>143415</td>
<td>0.8155105483724229</td>
<td>No Hit</td>
</tr>
<tr>
<td>AGAACATGAACGCTGATCTCCCAAGCAGTGGGAGGAGCCTGGGTCTCG</td>
<td>111584</td>
<td>0.6345077504066322</td>
<td>No Hit</td>
</tr>
<tr>
<td>AAAACGACTCTCGGCAACGGATACTCGGCTCTCGCATCGATGAAGACAG</td>
<td>111255</td>
<td>0.6326369351474214</td>
<td>No Hit</td>
</tr>
<tr>
<td>ATTCGATAGGTGTCAAGTGGAGTTGAGTGCATGATTGCAGTGAGGCTAGT</td>
<td>73682</td>
<td>0.41898300890326096</td>
<td>No Hit</td>
</tr>
<tr>
<td>GAAGGTCACGGCAGACGAGCCGTTTATCATTACGATAGGTTGCAAGGGGG</td>
<td>71661</td>
<td>0.4074908580252516</td>
<td>No Hit</td>
</tr>
<tr>
<td>GGATGGCATCATTACACAGCAGACTAATTGCCACGGATCCCATCGAAGACTCCGA</td>
<td>69548</td>
<td>0.3954755612388914</td>
<td>No Hit</td>
</tr>
<tr>
<td>ATATTCTGGTGTCTCTAGCGTAGGGAACACCAATCCATCCGAACCTT</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 *(always needed for RNA-seq alignment)*
   - **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
FASTQ trimming and adapter removal

Tools:
- cutadapt – https://code.google.com/p/cutadapt/
- trimmomatic – http://www.usadellab.org/cms/?page=trimmomatic
- FASTX-Toolkit – http://hannonlab.cshl.edu/fastx_toolkit/

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
    - “unteemplated” adapter sequences will be “soft clipped” (ignored)

```
  CACAAGTACAAATTATACAC
  TACATACACAAGTACAAATTATACACAGACATTAGTTCTTATCGCCCTGAA
```

(reference sequence)
Part 4: Alignment to a reference assembly

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

**core processes**

1. QC raw read sequences
   - experimental design
   - DNA/RNA isolation
   - library preparation
   - next-gen sequencing
   - delivery of raw reads

2. Has reference?
   - yes: map reads to reference
   - no: assembly (genome or transcriptome)

3. Alignment metrics & QC
   - reference assembly
   - fasta

4. Basic analysis
   - (e.g. coverage, genes)
   - bed, gff, vcf, etc.

5. Further analysis & significance determination
   - (e.g. FPKM, peak or variant calls)

6. Confident calls

**upstream processes**

**downstream processes**

- assembly (genome or transcriptome)
- metrics & QC
- delivery of raw reads
- core processes
- differential analysis
- annotation
- motif analysis
- custom analysis
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: \( P(\text{incorrectly mapped}) = 10^{-\text{mappingQuality}/10} \)
  - **alignment score** describes *fit*
    - reflects the correspondence between the read and the reference sequence

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

- Maps to 2 locations:
  - **low mapping quality**
  - Matches perfectly:
    - **high alignment score**
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>ATAGCTAATCCAAC</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.

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIX.MIXED.PIXIES.SIFT.SIXTY.PIXIE.DUST.BOXES</td>
<td>TEXYDST.E.IXIXIXXSSMPPS.B.E.S.EUSFXDIIIOIT</td>
</tr>
</tbody>
</table>

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)

- Alignment score = \sum
  - 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
ATTTGCAGATCGGATGAAGACGAA
ATTTGCAGATCGGATGTTGACTTT
ATTTGCAGATCGGATGAAGACG..AA
ATTTGCAGATCGGATGTTGACTTTAA
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 fragment size is reported in the alignment record
  - for both proper and discordant pairs
Alignment Workflow

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

Tools:
- FastQC, cutadapt
- bwa index, bowtie2-build
- bwa aln + bwa samse or sampe, bwa mem, or bowtie2
- samtools view
- samtools sort
- Picard MarkDuplicates
- samtools rmdup
- samtools index
- samtools flagstat
- samtools idxstat

Resources:
Obtaining/building a reference

- **What is a reference?**
  - assembled genomes or transcriptomes
    - Ensembl, UCSC, for eukaryotes
    - NCBI RefSeq or GenBank for prokaryotes/microbes (prefer RefSeq)
  - any set of named DNA sequences
    - names can be chromosomes, genes, etc. (technically referred to as **contigs**)

- **Building a reference index** (aligner-specific)
  - may take several hours to build
    - but you build each index once, use for multiple alignments
  - requires FASTA files (.fa, .fasta) containing DNA sequences
    - annotations (genome feature files, .gff) may also be used to build the index, but will definitely be needed for downstream analysis

```bash
>chrM  Mitochondrial Chromosome
GATCACAGGTCTATCACCCTATTAACCACCTCACGCGAGCTCTCCATGCAT
TTGGTATTTTCGTCTGGGGGTGTGCACGCGATAGCATTGCGAGACGCTG
GAGCCGGAGCACCCTATGTCGAGCAGTATCTGCTTGGATTCCTGCTCAT
```

**sequence name** line
- **always** starts with `>`
- followed by a **name** and other (optional) descriptive information
- one or more line(s) of **sequence characters**
- **never** starts with `>`
Alignment Workflow

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

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

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

4. **Convert SAM to BAM**

5. **Sort BAM by position**

6. **Handle duplicates (optional)**

7. **Index BAM**

8. **Alignment metrics & QC**

**Tools and Resources**

- Picard MarkDuplicates
- samtools view
- samtools sort
- samtools index
- samtools flagstat
- samtools idxstat
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’d) format
    - *much* smaller than SAM files
    - when sorted + 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 bit-wise FLAGs</td>
</tr>
<tr>
<td>2</td>
<td>FLAG</td>
<td>Int</td>
<td>[0,2^{16}-1]</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>RNAME</td>
<td>String</td>
<td>* <em>([-()+-&lt;&gt;=][!-]</em></td>
<td>Reference sequence NAME</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]+) * ([MIDNSHPX]=) )+</td>
</tr>
<tr>
<td>7</td>
<td>RNEXT</td>
<td>String</td>
<td>*</td>
<td>=</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>![?-A-]*</td>
<td>ASCII of Phred-scaled base QUALity+33</td>
</tr>
</tbody>
</table>

---

**Example**

- **SRR030257.264529**
  - Read name: NC_012967
  - Contig + start = locus
  - Insert size, if paired
  - **99** = 1521
  - **29** = 34M2S
  - **1564**

- **SRR030257.2669090**
  - Read name: NC_012967
  - Contig + start = locus
  - Insert size, if paired
  - **147** = 1521
  - **60** = 36M
  - **1458**

---

**Positive for plus strand reads**

| 79 |

**Negative for minus strand reads**

| -99 |
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:
- FastQC, cutadapt
- bwa aln + bwa samse or sampe
- bwa mem, or bowtie2
- samtools view
- samtools sort
- Picard MarkDuplicates
- samtools rmdup
- samtools index
- samtools flagstat
- samtools idxstat

References:
- http://broadinstitute.github.io/picard/
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
2. Build aligner-specific reference index
3. QC & trim raw reads
4. Align reads to reference
5. Convert SAM to BAM
6. Sort BAM by position
7. Handle duplicates (optional)
8. Index BAM
9. Alignment metrics & QC

Tools:
- FastQC, cutadapt
- BWA index
- Bowtie2
- SAMtools view
- SAMtools sort
- Picard MarkDuplicates
- Samtools rmdup
- Samtools index
- 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 and BRCF “pods”
    - align_bowtie2_illumina.sh, align_bwa_illumina.sh, trim_adapters.sh, etc.
    - TACC: shared project directory /work/projects/BiolTeam/common/script/
    - BRCF pods: read-only mount /mnt/bioi/
Other NGS Resources at UT

- CBRS training courses
  - Intro to NGS, RNAseq, many others
    https://research.utexas.edu/cbrs/cores/cbb/educational-resources/
  - “Summer School” (4 or 5 half-day sessions in June)
    - lots of hands-on, including w/TACC
  - Short courses (3-4 hour workshops)

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

- Bioinformatics consultants
  - Dennis Wylie, Dhivya Arasappan, Benni Goetz, Anna
  - BioITeam wiki – https://wikis.utexas.edu/display/bioiteam/

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