Differential gene expression analysis using RNA-seq

Applied Bioinformatics Core, July 2019

Friederike Dündar with Luce Skrabanek & Paul Zumbo
Day 2: Identifying the transcripts that were sequenced

1. FastQC results
2. Experimental Design
3. Reference genome & transcript annotation
4. Alignment
   - STAR
   - BAM/SAM files
5. QC of alignment step
FastQC & MultiQC

randomly selected 8 biological replicates for each condition

```
mkdir raw_reads_QC/fastqc_results

for GENOTYPE in WT SNF2
do
    for i in 1 2 5 6 13 21 25 28 # random selection
do
    echo Running FastQC for: ${GENOTYPE} Sample No ${i}
    # make a folder for every sample's FastQC results
    mkdir raw_reads_QC/fastqc_results/${GENOTYPE}_${i}
    # run FastQC
    ~/mat/software/FastQC/fastqc ~/precomputed/rawReads_yeast_Gierlinski/${GENOTYPE}_${i}/*stq.gz \
        -o raw_reads_QC/fastqc_results/${GENOTYPE}_${i} -q

done

cd raw_reads_QC/fastqc_results/

# run MultiQC to summarize all the FastQC results into one document
~/mat/software/anaconda2/bin/multiqc . --dirs --interactive
# --dirs will use the folder names as sample names in the output
```
Two basic questions of QC

- How successful was the actual **sequencing**?
  - consistently high base call confidence

- Did our **library pep** generate a **faithful representation** of the DNA/RNA molecules in our samples?
  - ideally, the entire universe of transcripts has been sufficiently sampled (diverse library)
  - no **contaminations** (rRNA, foreign DNA, adapters, primers, …)
  - no bias towards fragments of certain **GC contents**/sizes
  - no **degradation** [cannot be assessed without alignment]
Sequencing quality per cycle

Phred score

Noise/uncertainty = fluorophore intensity not as clear as expected
Physically localized error rates

A typical flowcell has 8 lanes.

Each lane contains two columns of tiles.

Each tile is imaged 4x per cycle.

Each column contains >50 tiles.

Per tile sequence quality

Quality per tile vs. position in read (bp)
Sequence composition

“normal” RNA-seq pattern
→ random hexamer priming not sufficiently random

highly irregular pattern
often indicative of adapter contamination
More QC details


• https://rtsf.natsci.msu.edu/genomics/tech-notes/fastqc-tutorial-and-faq

• https://sequencing.qcfail.com/

• https://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/
EXPERIMENTAL DESIGN

How to avoid spurious signals and drowning in noise
Why do we need replicates?

**Goal:** Identify differences in expression for every gene.

...and “differences” should preferably be due to our experiment, not noise!

“Samples are our windows to the population, and their statistics are used to estimate those of the population.”

Martin Krzywinski & Naomi Altman

```
testdat <- data.frame(exprs = rnorm(200),
                      condition = c("WT", "MUT"),
                      gene_name = "Rando1A")
```
Invest in replicates!

- **recommended:** 6 biological replicates per condition for DGE of strongly changing genes \((\text{logFC} \geq 2)\) [based on insights from the fairly simple yeast transcriptome]

The most effective way to improve detection of differential expression in low expression genes is to add more biological replicates, rather than adding more reads (see Rapaport et al., 2013)
Replicate types

**Technical replicates**
- RNA extraction
- library prep
- sequencing lane
- sequencing lane
- sequencing lane

**Biological replicates**
- RNA from an independent growth of cells/tissue
- RNA extraction
- library prep
- sequencing lane
- sequencing lane
- sequencing lane

also see course notes and Blainey et al. (2014) Nature Methods, 1(9) 879–880.
“Once we accounted for the batch effect (...), the comparative gene expression data no longer clustered by species, and instead, we observed a clear tendency for clustering by tissue.”

“Overall, our results indicate that there is considerable RNA expression diversity between humans and mice, well beyond what was described previously, likely reflecting the fundamental physiological differences between these two organisms.”
ENCODEx’s study design was not optimal

Most human samples were sequenced separately from the mouse samples:

<table>
<thead>
<tr>
<th>Tissue</th>
<th>Human</th>
<th>Mouse</th>
</tr>
</thead>
<tbody>
<tr>
<td>adipose</td>
<td>female</td>
<td>male</td>
</tr>
<tr>
<td>adrenal</td>
<td>male</td>
<td>female</td>
</tr>
<tr>
<td>brain</td>
<td>female</td>
<td>male</td>
</tr>
<tr>
<td>heart</td>
<td>male</td>
<td>male</td>
</tr>
<tr>
<td>kidney</td>
<td>male</td>
<td>female</td>
</tr>
<tr>
<td>liver</td>
<td>male</td>
<td>female</td>
</tr>
<tr>
<td>lung</td>
<td>female</td>
<td>female</td>
</tr>
<tr>
<td>ovary</td>
<td>female</td>
<td>female</td>
</tr>
<tr>
<td>pancreas</td>
<td>female</td>
<td>female</td>
</tr>
<tr>
<td>sigmoid colon</td>
<td>male</td>
<td>female</td>
</tr>
<tr>
<td>small bowel</td>
<td>female</td>
<td>female</td>
</tr>
<tr>
<td>spleen</td>
<td>female</td>
<td>male</td>
</tr>
<tr>
<td>testis</td>
<td>male</td>
<td>male</td>
</tr>
</tbody>
</table>

not all variables can be controlled for human data: deceased organ donors
mouse data: 10-week-old littermates

and that’s ok, but you’ve got to be mindful of these limitations when making bold claims

A very good read (including the reviews and comments) that discusses many scientific as well as ethical issues: https://f1000research.com/articles/4-121/v1

* not just ENCODE: see e.g. Leek et al. (2010) Nat Rev Gen 11(10) 733-739 or Jaffe & Irizarry (2014) Genome Biol 15(R31) 1–9
Avoiding bias

**Completely randomized design**

<table>
<thead>
<tr>
<th>STRESS</th>
<th>A</th>
<th>B</th>
<th>A</th>
<th>A</th>
<th>B</th>
<th>A</th>
<th>B</th>
<th>A</th>
<th>A</th>
<th>B</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIET</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

**Restricted randomized design**

<table>
<thead>
<tr>
<th>GENOTYPE</th>
<th>A</th>
<th>A</th>
<th>A</th>
<th>A</th>
<th>A</th>
<th>A</th>
<th>B</th>
<th>B</th>
<th>B</th>
<th>B</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIET</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

**Blocked & randomized design**

<table>
<thead>
<tr>
<th>GENOTYPE</th>
<th>A</th>
<th>A</th>
<th>B</th>
<th>B</th>
<th>A</th>
<th>A</th>
<th>B</th>
<th>B</th>
<th>A</th>
<th>A</th>
<th>B</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIET</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

**Keep calm and flip a coin**

Block what you can, randomize what you cannot.

What factors are of interest? Which ones might introduce noise? Which nuisance factors do you absolutely need to account for?

Typical RNA-seq set-up

• keep the **technical nuisance** factors (harvest date, RNA extraction kit, sequencing date…) to a **minimum**

• cover only as much of the **biological variation as needed** (but keep possible limitations for the final conclusions in mind)

Make sure the sequencing core **multiplexes** all samples!
How deep is deep enough?

for DGE (logFC~ 2) in mammals: 20 – 50 mio SR, 75 bp

Goals that require more, longer, and possibly paired-end reads:

- quantification of lowly expressed genes
- identification of genes with small changes between conditions
- investigation of alternative splicing/isoform quantification
- identification of novel transcripts, chimeric transcripts
- de novo transcriptome assembly

Remember: The addition of replicate samples provides substantially greater detection power of DE than increased sequence depth. (Rapaport et al., 2013)

https://www.encodeproject.org/
https://doi.org/10.1186/gb-2013-14-9-r95
Summary I

• RNA-seq analysis is not a completely solved issue – but **DE analysis on a gene level** is decently mature and the field seems to gravitate towards some sort of standard

• no analysis tool can enforce (or replace!) common sense and knowledge about the biology behind the experiment

• crap in, crap out

• more replicates are often better investments than more reads

• FastQC and multiQC are great tools that you should use
READ MAPPING
Finding out where the reads came from
Read mapping

Base calling & demultiplexing
Bustard/RTA/OLB, CASAVA

FASTQC ➔ Raw reads
.tif

RSeQC, QoRTs ➔ Aligned reads
.fastq
.sam/.bam

Read count table

Normalized read count table

List of fold changes & statistical values

Downstream analyses on DE genes

1. Theoretical background
2. Annotation files
3. Files with aligned reads: SAM/BAM
4. Looking at aligned reads
Different philosophies of transcript quantification

**alignment** followed by **counting** of reads overlapping with genes/exons

*e.g.* STAR + featureCounts

estimating expression levels of individual isoforms based on **alignment-free k-mer matching**

*salmon, kallisto*

Both approaches absolutely rely on excellent reference sequences.

---

**Query sequences**

<table>
<thead>
<tr>
<th>Target Sequence</th>
<th>5' ACTACTAGATTACTTACGGATCGAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Sequence</td>
<td>5' TACTCACGGATGAG</td>
</tr>
</tbody>
</table>

**Word size: 3**

<table>
<thead>
<tr>
<th>$W^x_3$</th>
<th>ATG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TGT</td>
</tr>
<tr>
<td></td>
<td>GTG</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$W^y_3$</th>
<th>CAT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ATG</td>
</tr>
<tr>
<td></td>
<td>TGT</td>
</tr>
<tr>
<td></td>
<td>GTG</td>
</tr>
</tbody>
</table>

**Union of two sets**

$W_3 = W^x_3 \cup W^y_3$

|CAT|ATG|TGT|GTG|

**Word counts**

<table>
<thead>
<tr>
<th>$c^x_3$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c^y_3$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Euclidean distance**

$\|c^x_3 - c^y_3\| = \sqrt{(0-1)^2 + (1-1)^2 + (2-1)^2 + (2-1)^2} = \sqrt{3} = 1.73$

---

Alignment = lining up the letters of two (or more) strings so that each letter in S1 either matches a gap or another letter in S2.

To find the best alignment, we need:

- scoring function for the edit distance
- efficient alignment-solving algorithm

edit distance
= number of changes that are needed to match S1 and S2

choices made by the programmer of a given tools

Needleman-Wunsch | Smith-Waterman | BLAST
Seed-and-extend strategy

(1) **generate** raw seed from a read

(2) **identify** a **match** for the seed

(3) **extend** the matched seed and do the local alignment through standard dynamic programming algorithms based on Needleman–Wunsch or Smith–Waterman algorithm

(algorithms: Smith-Waterman or Needleman-Wunsch)
Aligning short RNA-seq reads

Particular challenges of Illumina sequencing:

- the query sequences (= reads) are very short
- there are millions of them!
- cannot expect 100% exact matches
  - seq. errors
  - biological variation
  - reference errors
- RNA-seq: some cDNA fragments can only be aligned if one allows for gigantic gaps (= introns)
Aligning short RNA-seq reads

Spliced alignment tools usually need:

1) reference genome for the alignment
2) annotation to inform decisions about where to allow gaps in the alignment

greatest downside of alignment approach: it’s resource-intensive!

… and the result is not inherently quantitative (it’s just read coordinates, really)!
Pseudo-alignment = alignment-free $k$-mer matching

sequences are split into $k$-mers, which can then be represented as nodes

$k$-mers shared between the read and a transcript set

transcript set $k$-mers without matches in the read
Kallisto’s pseudoalignment

- gene
- isoforms

Read nodes with the same transcript “equivalence class”

All possible equivalence classes for a given read

Final transcript equivalence class for the given read

Bray et al. (2016). doi: 10.1038/nbt.3519
# Alignment vs. lightweight mapping

<table>
<thead>
<tr>
<th>Example workflow</th>
<th>Alignment</th>
<th>Pseudo-alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>STAR + featureCounts</td>
<td>salmon</td>
</tr>
</tbody>
</table>

**Read mapping based on:**

- **Alignment**: Where does a read match best?
- **Pseudo-alignment**: Which collection of unique k-mer’s does a given read match best?

**Reference needed:**

- **Alignment**: Genome sequence + exon boundaries
- **Pseudo-alignment**: cDNA sequences

**Mapping result**

- **Alignment**: Genome coordinates (BAM)
- **Pseudo-alignment**: Table of expression level estimates (txt)

**Expression quantification:**

- **Alignment**: Counting how many reads overlap a gene.
- **Pseudo-alignment**: Summing the values assigned to each collection of unique k-mers (equivalence class)

**Output:**

- **Alignment**: Read counts (integers)
- **Pseudo-alignment**: Estimated transcript abundances (numeric)

**Speed**

- **Alignment**: ++ & +++
- **Pseudo-alignment**: ++++
Read mapping

Base calling & demultiplexing
Bustard/RTA/OLB, CASAVA

FASTQC ➔ Raw reads

RSeQC, QoRTs ➔ Aligned reads

Read count table

Normalized read count table

List of fold changes & statistical values

Downstream analyses on DE genes

Mapping

STAR

1. Theoretical background
2. Reference & annotation files
3. Files with aligned reads: SAM/BAM
4. Looking at aligned reads
Reference sequences

- reference sequences (genome, cDNA, …) were originally produced with Sanger sequencing
- most reference sequences will undergo continuous refinement (→ “genome versions”)
- RefSeq & Ensembl are two pan-species databases with homogenous computational annotation workflows
- reference genomes are longer, but less ambiguous than reference transcriptome sequences!
Most individual RNA variations do not find their way into the reference sequences.

Zhao and Zhang had highly consistent or nearly identical expression levels, but this was clearly not the case. Although the majority of genes regardless of the choice of a gene model; however, this identical numbers of mapped reads for all common genes, those genes with zero counts. Ideally, we should get was added to the counts to avoid a logarithmic error for and y-axes represented log2(count + 1). For all genes, 1 21,598 common genes. The overall correlation between gene quantification results, we focused on this set of annotations.

21,598 common genes are shared by all three gene models. The Venn diagram in Figure 4 showed the overlap and intersection among RefGene, UCSC, and Ensembl. Considering that annotations are more or less incomplete in these databases, we only focused on common gene models originating from these databases. Therefore, we mapped those database-specific junction reads were mapped to more than one location for which the number of reads mapped to them was 0 in one gene model, but many in others.

As shown in Figure 5, there were many genes with expression levels differing by 5% or more was only a few extreme or representative cases to provide possible explanations. In the liver sample, the expression levels for PIK3CA gene definition in both Ensembl and RefGene, were summarized in Table 2 (read length = 75 bp). Compared with Ensembl, there are three isoforms for PIK3CA, and the longest isoform is ENST00000263967. The total length of PtdIns4P, and PtdIns(4,5)P2. In the liver sample, there were a significant number of genes whose quantification was added to the counts to avoid a logarithmic error for and y-axes represented log2(count + 1). For all genes, 1 21,598 common genes. The overall correlation between gene quantification results, we focused on this set of annotations.

Figure 4 showed the overlap and intersection among RefGene, UCSC, and Ensembl. Considering that annotations are more or less incomplete in these databases, we only focused on common gene models originating from these databases. Therefore, we mapped those database-specific junction reads were mapped to more than one location for which the number of reads mapped to them was 0 in one gene model, but many in others.

As shown in Figure 5, there were many genes with expression levels differing by 5% or more was only a few extreme or representative cases to provide possible explanations. In the liver sample, the expression levels for PIK3CA gene definition in both Ensembl and RefGene, were summarized in Table 2 (read length = 75 bp). Compared with Ensembl, there are three isoforms for PIK3CA, and the longest isoform is ENST00000263967. The total length of PtdIns4P, and PtdIns(4,5)P2. In the liver sample, there were a significant number of genes whose quantification

---

### Annotation: defining transcript structures

- **Annotation is dynamic!** (sequence, coordinates, types of elements)
- **Automated vs. manual curation (“evidence-based”)**

[Diagram of Genome, Chromosome, Gene, Intergenic region]

[Table showing overlaps between different gene annotations]

**RefSeq** ncbi.nlm.nih.gov/refseq
**UCSC Known Genes** genome.ucsc.edu
**Ensembl/Gencode** gencodegenes.org

1/3 protein-coding genes
> 17,000 non-coding RNAs
> 15,000 pseudogenes
## Integrative genome annotation

<table>
<thead>
<tr>
<th>Species</th>
<th>Ensembl</th>
<th>RefSeq</th>
<th>Specialized db</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Human</strong></td>
<td>206601</td>
<td>135907</td>
<td>206694</td>
</tr>
<tr>
<td></td>
<td>58735</td>
<td>38711</td>
<td>58721</td>
</tr>
<tr>
<td><strong>Mouse</strong></td>
<td>138930</td>
<td>103177</td>
<td>138835</td>
</tr>
<tr>
<td></td>
<td>54838</td>
<td>36035</td>
<td>54752</td>
</tr>
<tr>
<td><strong>Worm</strong></td>
<td>61109</td>
<td>44377</td>
<td>61574</td>
</tr>
<tr>
<td></td>
<td>46778</td>
<td>28981</td>
<td>47269</td>
</tr>
<tr>
<td><strong>Fly</strong></td>
<td>34767</td>
<td>34114</td>
<td>35359</td>
</tr>
<tr>
<td></td>
<td>17737</td>
<td>17101</td>
<td>17773</td>
</tr>
<tr>
<td><strong>Arabidopsis</strong></td>
<td>55398</td>
<td>48030</td>
<td>48359</td>
</tr>
<tr>
<td></td>
<td>34218</td>
<td>22132</td>
<td>27655</td>
</tr>
<tr>
<td><strong>Yeast</strong></td>
<td>7127</td>
<td>12236</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>7024</td>
<td>5123</td>
<td>NA</td>
</tr>
</tbody>
</table>

## Direct RNA-seq assembly

<table>
<thead>
<tr>
<th>Method</th>
<th>Transcripts</th>
<th>Genes</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mitranscriptome</strong></td>
<td>384066</td>
<td>91013</td>
<td>4.22</td>
</tr>
<tr>
<td><strong>Big transcriptome</strong></td>
<td>338859</td>
<td>46634</td>
<td>7.26</td>
</tr>
<tr>
<td><strong>CHESS</strong></td>
<td>323258</td>
<td>42611</td>
<td>7.59</td>
</tr>
</tbody>
</table>
Which annotation should one use?

“More sensitive annotations, such as Ensembl (...) should be preferred over more specific annotations, such as RefSeq (...) if the aim is to obtain accurate expression estimates.“

Janes et al. (Briefings in Bioinformatics, 2015). doi: 10.1093/bib/bbv007

“We observe that RefSeq Genes produces the most accurate fold-change measures with respect to a ground truth of RT-qPCR gene expression estimates.“

Wu et al. (BMC Bioinfo, 2013). doi: 10.1186/1471-2105-14-S11-S8

“In practice, there is no simple answer to this question, and it depends on the purpose of the analysis. (...) When choosing an annotation database, researchers should keep in mind that no database is perfect and some gene annotations might be inaccurate or entirely wrong.”

Gene models can differ dramatically

[Diagram showing gene definitions in Ensembl vs. RefSeq]

1. **ENSEMBL**
   - 0 unambiguous reads for LUZP6;
   - x number of reads for MTPN

2. **RefSeq**
   - 0 reads for either
   - or the same number of reads for both genes

Pay attention to the source as well as to the version of the genome/annotation build!
Storing annotation information

- representing genome coordinates + description/name
- various formats (all are plain text files): GFF2, GFF3, GTF, BED, SAF...

GTF ("GFF2.5")
1. reference coordinate
2. source
3. annotation type
4. start position
5. end position
6. score
7. strand
8. frame/phase
9. attributes: <TYPE VALUE>; <TYPE VALUE>; <TYPE VALUE>

see the course notes for details
0 vs. 1 based conventions

**one-based, fully-closed**

ATG location: 7 - 9 or [7,9]
Cut site: 11-12 or (11,12)
Interval length = stop - start + 1

**zero-based, half open**

ATG location: 6 - 9 or [6,9)
Cut site: 11-11 or [11,11)
Interval length = stop - start

---

GFF format

BED format

---

http://alternateallele.blogspot.de/2012/03/genome-coordinate-conventions.html

http://alternateallele.blogspot.com/2012/03/genome-coordinate-cheat-sheet.html
Spliced Transcriptome Alignment to Reference (STAR)

- accurate & sensitive
- very fast
- memory intensive!
(use it on the server!)

- MMP = maximal mappable prefix
  (aka maximum matching portion)
- reads are split when a continuous alignment is not possible
- the remaining unmappable portion is then aligned again
- finally, aligned portions of the original full-length reads are stitched together

Engström et al. (2013) Nature Methods, 10(12), 1185-1191. doi:10.1038/nmeth.2722
**STAR spliced alignment**

- accurate & sensitive
- very fast
- memory intensive!

**Spliced alignment programs**

- BAGET ann
- GEM ann
- GEM cons
- GEM cons ann
- GSNAP
- GSNAP ann
- GSTRUCT
- GSTRUCT ann
- MapSplice
- MapSplice ann
- PALMapper
- PALMapper ann
- PALMapper cons
- PALMapper cons ann
- PASS
- PASS cons
- ReadsMap
- SMALT
- STAR 1-pass
- STAR 1-pass ann
- STAR 2-pass
- STAR 2-pass ann
- TopHat1
- TopHat1 ann
- TopHat2
- TopHat2 ann

% mapped fragments

**STAR has myriad options! Tune them to meet your needs**

Current Protocols in Bioinformatics (Sept 2015)
DOI: 10.1002/0471250953.bi1114s51
and
STARmanual.pdf
2 main STAR modules

1. **generate genome index**
   - `--runMode genomeGenerate`
   - `--genomeFastaFiles sacCer3.fa`
   - `--sjdbGTFfile sacCer3.gtf`
   - needs to be done just 1x per transcriptome!

2. **align**
   - 2.1. **align to reference & identify novel splice junctions**
   - 2.2 **re-run** alignment including the novel splice junctions

Let's align the reads for WT_1!
Storing aligned reads: SAM/BAM

<table>
<thead>
<tr>
<th>HD</th>
<th>VN:</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQ</td>
<td>SN:</td>
</tr>
<tr>
<td>RG</td>
<td>ID:</td>
</tr>
<tr>
<td>PG</td>
<td>ID:</td>
</tr>
<tr>
<td>CO</td>
<td></td>
</tr>
</tbody>
</table>

### Header Section
The header section contains general information about the file.

**Row Description**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>QNAME</td>
</tr>
<tr>
<td>2</td>
<td>FLAG</td>
</tr>
<tr>
<td>3</td>
<td>RNAME</td>
</tr>
<tr>
<td>4</td>
<td>POS</td>
</tr>
<tr>
<td>5</td>
<td>MAPQ</td>
</tr>
<tr>
<td>6</td>
<td>CIGAR</td>
</tr>
<tr>
<td>7</td>
<td>RNEXT</td>
</tr>
<tr>
<td>8</td>
<td>PNEXT</td>
</tr>
<tr>
<td>9</td>
<td>TLEN</td>
</tr>
<tr>
<td>10</td>
<td>SEQ</td>
</tr>
<tr>
<td>11</td>
<td>QUAL</td>
</tr>
<tr>
<td>&gt;11</td>
<td>OPT</td>
</tr>
</tbody>
</table>

**Column Details**

- **QNAME**: Query name.
- **FLAG**: Flags indicating various conditions.
- **RNAME**: Reference name.
- **POS**: Position.
- **MAPQ**: Mapping quality.
- **CIGAR**: Cigar string indicating insertions, deletions, and matching pairs.
- **RNEXT**: Reference next name.
- **PNEXT**: Pair next name.
- **TLEN**: Template length.
- **SEQ**: Sequence.
- **QUAL**: Quality score.
- **OPT**: Optional information.

**Example Row**

<table>
<thead>
<tr>
<th>QNAME</th>
<th>FLAG</th>
<th>RNAME</th>
<th>POS</th>
<th>MAPQ</th>
<th>CIGAR</th>
<th>RNEXT</th>
<th>PNEXT</th>
<th>TLEN</th>
<th>SEQ</th>
<th>QUAL</th>
<th>OPT</th>
</tr>
</thead>
</table>

**Alignment Section**

1 line per locus.

<table>
<thead>
<tr>
<th>QNAME</th>
<th>FLAG</th>
<th>RNAME</th>
<th>POS</th>
<th>MAPQ</th>
<th>CIGAR</th>
<th>RNEXT</th>
<th>PNEXT</th>
<th>TLEN</th>
<th>SEQ</th>
<th>QUAL</th>
<th>OPT</th>
</tr>
</thead>
</table>

**Alignment Format**

- **AS**: Alignment score.
- **BC**: Base quality score.
- **NH**: Number of hard clips.
- **NM**: Number of soft clips.
- **Z**: Additional identifier.
- **H**: Hard clipping.

**Example Row**

```
QNAME FLAG RNAME POS MAPQ CIGAR RNEXT PNEXT TLEN SEQ QUAL OPT
QNAME FLAG RNAME POS MAPQ CIGAR RNEXT PNEXT TLEN SEQ QUAL OPT
QNAME FLAG RNAME POS MAPQ CIGAR RNEXT PNEXT TLEN SEQ QUAL OPT
QNAME FLAG RNAME POS MAPQ CIGAR RNEXT PNEXT TLEN SEQ QUAL OPT
```
Storing aligned reads: SAM/BAM

2nd field: binary FLAG

<table>
<thead>
<tr>
<th>Binary (Decimal)</th>
<th>Hex</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000001 (1)</td>
<td>0x1</td>
<td>Is the read paired?</td>
</tr>
<tr>
<td>00000000010 (2)</td>
<td>0x2</td>
<td>Are both reads in a pair mapped “properly” (i.e., in the correct orientation with respect to one another)?</td>
</tr>
<tr>
<td>00000000100 (4)</td>
<td>0x4</td>
<td>Is the read itself unmapped?</td>
</tr>
<tr>
<td>00000001000 (8)</td>
<td>0x8</td>
<td>Is the mate read unmapped?</td>
</tr>
<tr>
<td>00000100000 (16)</td>
<td>0x10</td>
<td>Has the read been mapped to the reverse strand?</td>
</tr>
<tr>
<td>00001000000 (32)</td>
<td>0x20</td>
<td>Has the mate read been mapped to the reverse strand?</td>
</tr>
<tr>
<td>00010000000 (64)</td>
<td>0x40</td>
<td>Is the read the first read in a pair?</td>
</tr>
<tr>
<td>00100000000 (128)</td>
<td>0x80</td>
<td>Is the read the second read in a pair?</td>
</tr>
<tr>
<td>01000000000 (256)</td>
<td>0x100</td>
<td>Is the alignment not primary? (A read with split matches may have multiple primary alignment records.)</td>
</tr>
<tr>
<td>10000000000 (512)</td>
<td>0x200</td>
<td>Does the read fail platform/vendor quality checks?</td>
</tr>
<tr>
<td>10000000000 (1024)</td>
<td>0x400</td>
<td>Is the read a PCR or optical duplicate?</td>
</tr>
</tbody>
</table>

most common FLAGS for SR: 0; 4; 16
Storing aligned reads: SAM/BAM

6th field: CIGAR string – which hoops did the aligner have to jump through to align the read to the genome locus that it thought was the best fit?

M  alignment (match or mismatch!!)
I (N)  insertion (large insertions)
D  deletion
S/H  clipping

spliced out introns = sequences are missing in the read, i.e., they need to be inserted in order to align the read to the genome

<table>
<thead>
<tr>
<th>Reference sequence with aligned reads</th>
<th>CIGAR string</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>C T G C A T G T T A G T A A * * G A T A G C T G T G C T A</td>
<td>1M2I4M1D3M</td>
<td>Insertion &amp; Deletion</td>
</tr>
<tr>
<td>A A G G A T A * C T G</td>
<td>5M1P1I4M</td>
<td>Padding &amp; Insertion</td>
</tr>
<tr>
<td>G A T A A * G G A T A</td>
<td>5M15N5M</td>
<td>Spliced read</td>
</tr>
<tr>
<td>T G T T A</td>
<td>3S8M</td>
<td>Soft clipping</td>
</tr>
<tr>
<td>a a a C A T G T T A G</td>
<td>3H8M</td>
<td>Hard clipping</td>
</tr>
</tbody>
</table>
Storing aligned reads: SAM/BAM

after 11\textsuperscript{th} field: OPTIONAL information

\textbf{AS:i} Alignment score
\textbf{BC:Z} Barcode sequence
\textbf{HI:i} Query is \textit{i}-th hit stored in the file
\textbf{NH:i} Number of reported alignments for the query sequence
\textbf{NM:i} Edit distance of the query to the reference
\textbf{MD:Z} String that contains the exact positions of mismatches (should complement the CIGAR string)
\textbf{RG:Z} Read group (should match the entry after ID if @RG is present in the header.

\hspace{1cm} \begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|}
\hline
QNAME & FLAG & RNAME & POS & MAPQ & CIGAR & RNEXT & PNEXT & TLEN & SEQ & QUAL & OPT \\
\hline
\end{tabular}

\text{tags are not standardized!}

\hspace{1cm} \begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|}
\hline
\text{TAG} & \text{TYPE} & \text{VALUE} \\
\hline
\end{tabular}

The myriad information stored within the alignment files allow you to focus on virtually any subset of read alignments that you may be interested in. \texttt{samtools view} has many options that directly interpret some of the mandatory fields of its alignment section (Table 3), such as the mapping quality, the location and the \texttt{FLAG} field values.

To see all the operations that can be done using \texttt{samtools}, type \texttt{samtools --help}.
Basic QC of aligned reads

How many reads were aligned? What were reasons for lack of alignment?
Do you have enough paired mates (for PE sequencing)?

- aligner output (e.g., Log.final.out)
- samtools flagstat
- RSeQC’s bam_stat
- QoRTs

(almost) all of these can be summarized using MultiQC!

→ Section 3.4.1 of the course notes
Integrative Genomics Viewer (IGV) (Version 2.3)

Install IGV

Options for installing and running IGV:

1. (Mac only) Download and run the Mac application; or
2. (Windows) Download and run the self-extracting archive; or
3. (All systems) Use the Java Web Start buttons (Mac users: see below for limitations); or
4. (All systems) Download the binary distribution and run IGV from the command line.

Note: IGV 2.3.x requires Java 7. Users with Java 6 (JRE 1.6) should first try to upgrade Java to the latest version. If this is not possible you will need to run a 2.2.x version available in the archive.

Mac

Download and unzip the Mac App archive, then double-click the IGV application to run it. The application can be moved to the "Applications" folder, or anywhere else.
Integrative Genomics Viewer

- load BAM file(s) from URL (“File” -> “Open URL…”):
  - http://chagall.med.cornell.edu/RNASEQcourse/
  - http://www.trii.org/courses/rnaseq.html

- take a snapshot of the reads around gene YPL198W

http://software.broadinstitute.org/software/igv/Sashimi

starting with IGV 2.3, Sashimi plots can easily be created

Typical biases of RNA-seq

- **lack of gene diversity:**
  - dominance of rRNAs, tRNAs or other highly abundant transcripts

- **read distribution**
  - high intron coverage: incomplete poly(A) enrichment
  - many intergenic reads: gDNA contamination

- **gene body coverage**
  - 3’ bias: RNA degradation + poly(A) enrichment

[Visualization of gene body coverage and read distribution]

Different protocols have different gene body coverage bias

Lahens et al. (2014) Genome Biology 15:R86

geneBody_coverage.py
removing rRNAs

Can be done at virtually every step of the analysis. Choose the version that makes most sense to you.

  - input: reads in fastq file + rRNA sequences
  - will extract those reads that do not match to the rRNA sequences

- make a **“genome” index for rRNAs only** (and perhaps tRNAs), then align your reads and only use those that do not map for the next round of alignment

- do your alignment and counting as is, simply **ignore the rRNA genes** in your subsequent downstream analysis
QC recap

- **raw reads QC**
  - adapter/primer/other contaminating and over-represented sequences
  - sequencing quality
  - GC distributions
  - duplication levels

- **aligned reads QC**
  - % (uniquely) aligned reads
  - % exonic vs. intronic/intergenic
  - gene diversity
  - gene body coverage

aligner’s log files
samtools flagstat
RSeQC
QoRTs
...
summarize with MultiQC!

FastQC
(QoRTs)
Summary

• aligning unspliced reads is not too difficult, but it still takes a long time (depending on the size of the genome)

• spliced reads are quite tricky, and identifying novel splice junctions is error-prone and far from being solved

• the file format for storing aligned reads (SAM/BAM) is fairly standardized, but the optional fields (and how alignment tools interpret some of the mandatory entries) leave lots of room for variability

• the file format(s) for storing genome annotation (e.g. genes, transcripts) tend to be even stricter defined and even less well followed (aka it’s a mess!)

• historically, samtools are the most widely used tools when it comes to exploring and manipulating SAM/BAM files (although there are alternatives, e.g. bamtools)

• QC of aligned read files is at least as important as QC of the raw reads, if not more so!