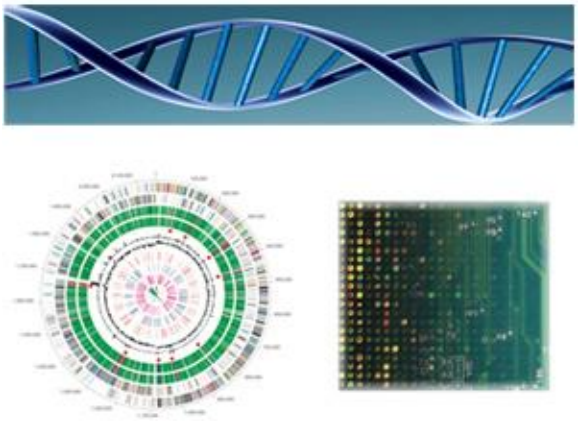
Bioinformatics Short Course: NGS QC and RNAseq



Integrative Bioinformatics

July 10 2013

Sara Grimm, PhD
grimmsa@niehs.nih.gov

NIH National Institute of Environmental Health Sciences

Module Overview

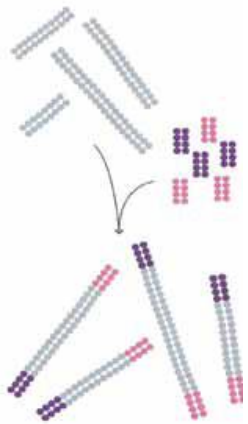
- Background on NGS [next-generation sequencing].
- Basic QC of NGS data.
- RNAseq analysis.
    - preparing input data
    - mapping reads
    - quantification of genomic features
    - visualization of data
    - comparing samples

Topic 1
NGS Background Info

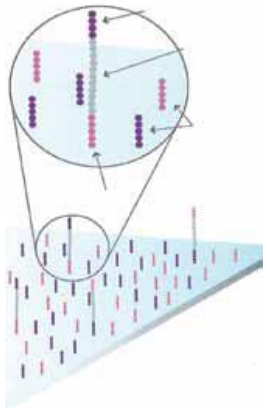
Illumina Sequencing Technology

http://www.illumina.com/Documents/products/techspotlights/techspotlight_sequencing.pdf

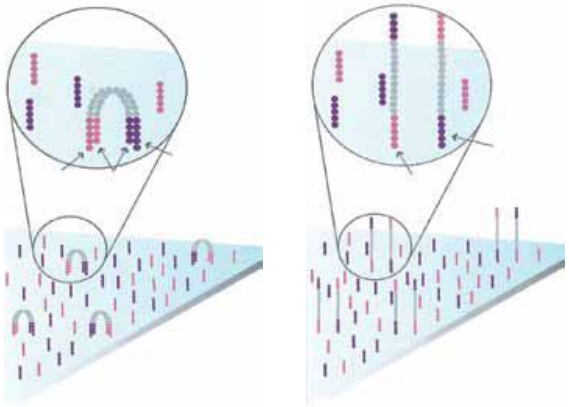
Prepare sequence fragments. Ligate adapters.



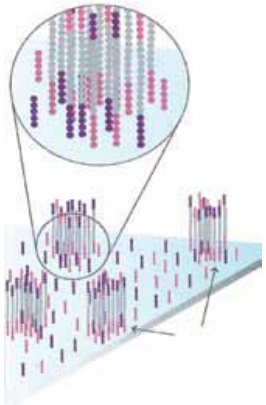
Attach fragments to surface of flow cell.



Bridge amplification. Denature DS molecule. Repeat.

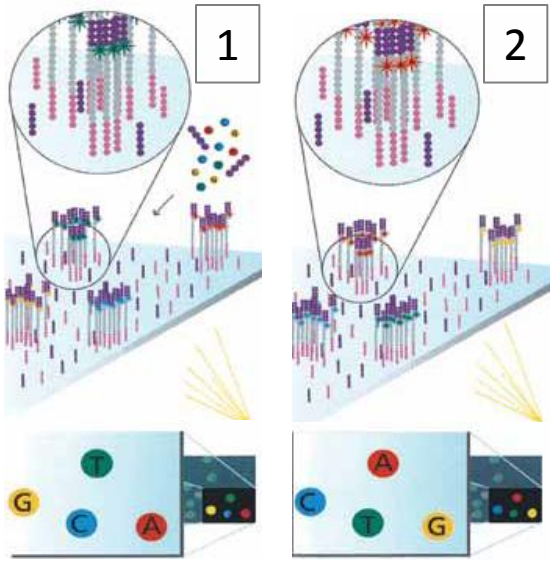


Dense clusters of identical fragments are generated.

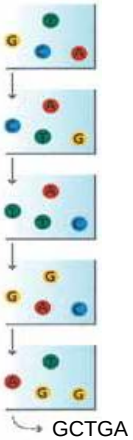


Add labeled reversible terminators. Emitted fluorescence identifies the incorporated base. Cleave fluorescent dye. Repeat.

[Number of cycles = read length.]

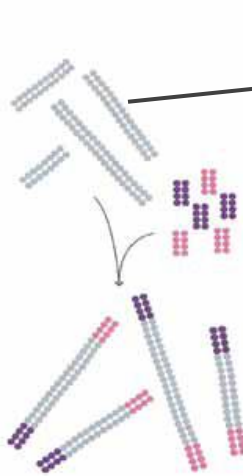


Images converted to “reads”.



NGS Data Basics: Single-end vs Paired-end Data

Prepare sequence fragments. Ligate adapters.



DNA/RNA fragment of known length

Single-end (SE) sequencing.

Paired-end (PE) sequencing.

Illumina Sequencing Output

	HiSeq 2000	GA IIx	MiSeq
Number of Lanes	16	8	1
Max. Output	600 Gb	95 Gb	8.5 Gb
Max. Single-End Reads	3 billion [187 million / lane]	320 million [40 million / lane]	17 million
Max. Paired-End Reads	6 billion [374 million / lane]	640 million [80 million / lane]	34 million
Max. Read Length	2 x 100	2 x 150	2 x 250

** Data from http://www.illumina.com/*

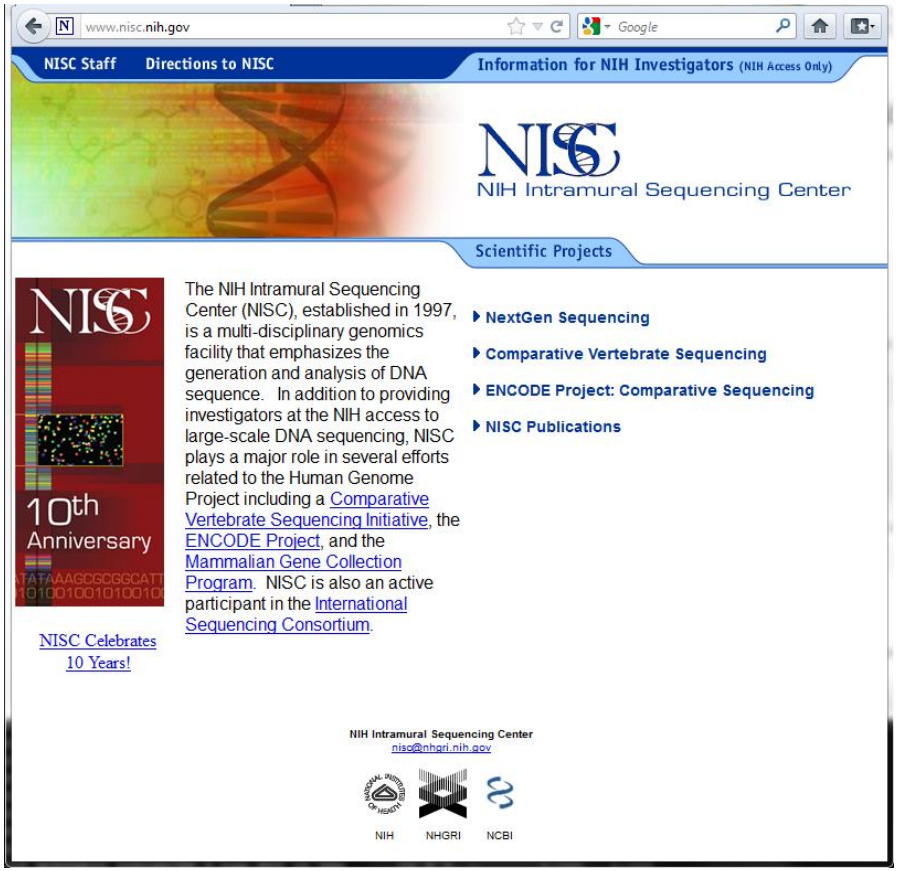
Sequencing @ NISC

Currently, sequencing from NISC comes in two standard options...

a) single-end 36mers on the GAIIx
   * 25-35 million reads

b) paired-end 101mers on the HiSeq
   * 175-225 million read pairs
     [350-450 million reads]

http://www.nisc.nih.gov/



Sequencing @ NIEHS

a) MiSeq: single-end (50, 300, or 500mers) or paired-end (25, 150, or 250mers)
   * 10-20 million reads

NGS Data Basics: FASTQ format, SE data

```
@C060CACXX:1:2108:04435:81967
AGAGAATGGTAG
+
?@@DDDFFHFFF
@C060CACXX:1
GTGCATTCTTAT
+
CCCFFFFFHHH
@C060CACXX:1
CTCCTTTCCCAT
+
==>AA@?;?++@<=?AC>BB4,A7,,3?A>4+???A<@BBBA7);*111*?0?3;=?A>A
@C060CACXX:1:1305:16126:134486
ATCTATTCCTGAACAGGTCAATTTTAATGACTGATTCTTCAATCCGTGGTGGTCGAGATG
+
;>=AAAAABB+@=@C3+?++<,,33<=C<+?77+*:=7*1?A?=3?0:0=A<A3(<AA##
@C060CACXX:1:1308:04529:41884
ATTTGCCATCCCTGCATTGTGCGTGGTTTTCAGCAGCTTTTTAACAGGTGTTGTTTTTAT
+
@@<DDDEAFHHFDIGEEGGE9FGHHIA@FGIIGIIGIIJJJJIIIIEHDDBFFBCGHGII
@C060CACXX:1:2202:06955:98871
CTGAGATCTTCTTTAATTTCTTTCTTCAGGGACTTGAAGTTTTTATCATACAGATCTTTC
+
BCCDFFFFHHHHHJJJJJJJJJJIJJJIJJJIIJJJGIIFIJJJJJJJJJJIJJJJJJIJ
@C060CACXX:1:1105:15276:91210
TAGGAATCAGCGTGAGCTGTATTCTGACGGAGAATCTCTTCTGGTACCAGAAGGTTTGGA
+
?7?>BDD:C3:02@+AE2<3AEEDF++<))?D?DD4BDB9DDIIDBDD49DB;8.48@5@
@C060CACXX:1:1301:16367:35650
CGCTCTCCAAGCTCCTCCTCCTGGCCCTCAGCTTCTGTGGCTTTCTGGTCTTCACCAACC
+
==<;A8A7+?A7?CB9AAACA++++2<?)5@3*1????*0:?=>**00/*9AA43))==A
@C060CACXX:1:1205:17708:111304
CTGGTAGTAAAGTAGCTGCATGGAGTTCACCTGCAGTTCGTGCTGCTTGGCGCCGACCCA
+
?@@DABB=CC<,C:ACG4CFE4@E;+<?+<C3CDCFF?91::)0:?<93BG(7;;''58(
@C060CACXX:1:1208:13509:106734
GCTTTGTGGTCTTCACCAACCTTTCTCTGCAGAACAACACCATAGGCACCTATCAGCTGG
+
@CCFFFDFHFHHHJIJIJJJJJJJJJJIJIIJJJJIIJJJJEHIIJIGIIJJJJJJJIHJG
@C060CACXX:1:1101:03034:113094
ATTCTCCGTCAGAATACAGCTCACGCTGATTCCTATTACTGTAGGTGTAATCCTAAATTC
+
@CCFFFFFHHHFHIIIJIHIIIJJIIHIJEIJJGJBHGIGGDDFCDHEFFCIBGICHIIG
.
.
.
.
```

```
@C060CACXX:1:1305:16126:134486
ATCTATTCCTGAACAGGTCAATTTTAATGACTGATTCTTCAATCCGTGGTGGTCGAGATG
+
;>=AAAAABB+@=@C3+?++<,,33<=C<+?77+*:=7*1?A?=3?0:0=A<A3(<AA##
```

Standard format is 4 lines per read:

1. Unique read identifier.
2. Read sequence.
3. Either read identifier again or a placeholder like '+'.
4. Phred-like base quality scores [Q:0-40].

   $Q = -10 \log_{10}(e)$, where e is the estimated probability of a wrong base. So the probability that a base call is an error is…

   * 0.01% if Q=40
   * 0.1% if Q=30
   * 1% if Q=20
   * 10% if Q=10

NGS Data Basics: FASTQ format, PE data

```
@C060CACXX:1:2108:04435:81967/1
AGAGAATGGTACAGGTACCAACAACATGCCATATGCATAGAGCAGCACAGAGCAACATAA
+
?@@DDDFFHFFFHJJEHIJIJIGHHHIJJIJJJJJJ@HGHGICBFGCHIECGGGDHACBC
@C060CACXX:1:1103:08674:67296/1
GTGCATTCTTATTTTATAATATTGACTCTATGACTCAAAAATTACAAGTGTTTATAACCC
+
CCCFFFFFHHHGHJIGIIGIGHIGIJJJIJJJIIJIJJJJJJJJJJIJEGGIIJIGIICH
ATTTGCCATCCCTGCATTGTGCGTGGTTTTCAGCAGCTTTTTAACAGGTGTTGTTTTTAT
+
@@<DDDEAFHHFDIGEEGGE9FGHHIA@FGIIGIIGIIJJJJIIIIEHDDBFFBCGHGII
@C060CACX
CTGAGATCT
+
BCCDFFFFH
@C060CACX
TAGGAATCA
+
?7?>BDD:C
@C060CACX
CGCTCTCCAAGCTCCTCCTCCTGGCCCTCAGCTTCTGTGGCTTTCTGGTCTTCACCAACC
+
==<;A8A7+?A7?CB9AAACA++++2<?)5@3*1????*0:?=>**00/*9AA43))==A
@C060CACXX:1:1205:17708:111304/1
CTGGTAGTAAAGTAGCTGCATGGAGTTCACCTGCAGTTCGTGCTGCTTGGCGCCGACCCA
+
?@@DABB=CC<,C:ACG4CFE4@E;+<?+<C3CDCFF?91::)0:?<93BG(7;;''58(
@C060CACXX:1:1208:13509:106734/1
GCTTTGTGGTCTTCACCAACCTTTCTCTGCAGAACAACACCATAGGCACCTATCAGCTGG
+
@CCFFFDFHFHHHJIJIJJJJJJJJJJIJIIJJJJIIJJJJEHIIJIGIIJJJJJJJIHJG
@C060CACXX:1:1101:03034:113094/1
ATTCTCCGTCAGAATACAGCTCACGCTGATTCCTATTACTGTAGGTGTAATCCTAAATTC
+
@CCFFFFFHHHFHIIIJIHIIIJJIIHIJEIJJGJBHGIGGDDFCDHEFFCIBGICHIIG
.
.
.
.
```

```
@C060CACXX:1:2108:04435:81967/2
GGGAAATAGTTATTTTAGGAAGTAGAAGATTTTTCTCTTTGTGTCTGAGTCTTTCATTTG
+
??@DDBDEHF>,C:C@EFBCFHG>HHBDGGHD@<EHGGIJJEB1?F4*:BDGG9DGGI??
@C060CACXX:1:1103:08674:67296/2
GTTTTTATACCATTTCTAACACAACATCTTTGCAACAGAAGAATGTGGAATGGTGTTTCT
+
@CCFFFFDHHAFHIIJIHIJJIDIIIGGHIJJEIGIIJHEHIGGIFGIJIFFHBFGHIIG
CATGAGTT
(?DAG>BI
GTTTTTAT
#######
ATCTTATTCCTGAACAGGTCAATTTTAATGACTGATTCTTCAATCCGTGGTGGTCGAGAT
+
?B@+4=BDFFHBHGB<E@<+3A?CFBE39<?2ACDGC>DF?CDDDF:FBDDF?@F(<6@
AGTAAAAGTAGCTGCATGGAGTTCACCTGCAGGTCGTGCTGCTTGGCTCCGACCCACACT
+
+:+4+2=A22:+2A+A2A?<A:+<<CB9+<C?)1*:0)?B?B>DD)9*90?:;-;(;(;A
@C060CACXX:1:1205:17708:111304/2
GCTTTGTGGGCTTCACCAACCTTTCTCTGCAGAACAACACTATAGGCACCTATCAGCTGG
+
+:++AD22C)1<CAFDGF@G:E<+924C*91**1:3933B***9B*0*97?383BFH)))
@C060CACXX:1:1208:13509:106734/2
GCAGGCATGGCAGAAGACATGGGGGCCTGGTAGTAAAGTAGCTGCATGGAGTTCACCTGC
+
BBC+A@DDHFHHFIGIBGGIHJIGHJIIHJ?DGBDGAGBDFGIGIIIGHDCGHIIHCHFH
@C060CACXX:1:1101:03034:113094/2
GATAAGTTCACCATGAAAACGATTATTCCAGACAGCAGGACCATAAGCAAAGCAGAAACT
+
=?B=A=2A=C:CD++<CF++333<2+A+AE?9)1):C1)0)?F**900?BF3?F.8BF)/
.
.
.
.
```


```
@C060CACXX:1:2108:04435:81967/1
AGAGAATGGTACAGGTACCAACAACATGCCATATGCATAGAGCAGCACAGAGCAACATAA
+
?@@DDDFFHFFFHJJEHIJIJIGHHHIJJIJJJJJJ@HGHGICBFGCHIECGGGDHACBC
```

```
@C060CACXX:1:2108:04435:81967/2
GGGAAATAGTTATTTTAGGAAGTAGAAGATTTTTCTCTTTGTGTCTGAGTCTTTCATTTG
+
??@DDBDEHF>,C:C@EFBCFHG>HHBDGGHD@<EHGGIJJEB1?F4*:BDGG9DGGI??
```

Topic 2
NGS QC

NGS QC: Verify the quality of your data.

These apply to all NGS data (not just RNAseq). Examples of what to look for…

* sequence bias of reads relative to reference genome
* sequence bias across length of a read
* presence of sequencing contaminant, usually the primer/adapter
* low uniqueness of reported reads
* low base quality scores (i.e. low confidence of correct base calls)
* low read count

Some of these problems can be worked around to salvage some of the data.

Others indicate that the lane is bad & must be re-run (or a new library is needed).

There are a number of available NGS QC tools. My personal favorite is FastQC.

http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

NGS QC: Sequence bias relative to reference genome.

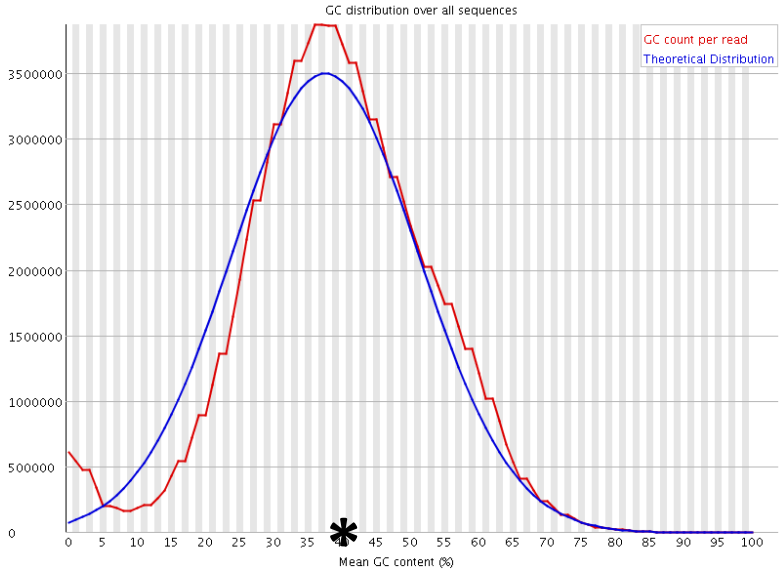
This plot shows the distribution of GC content per read for all reads in a lane.

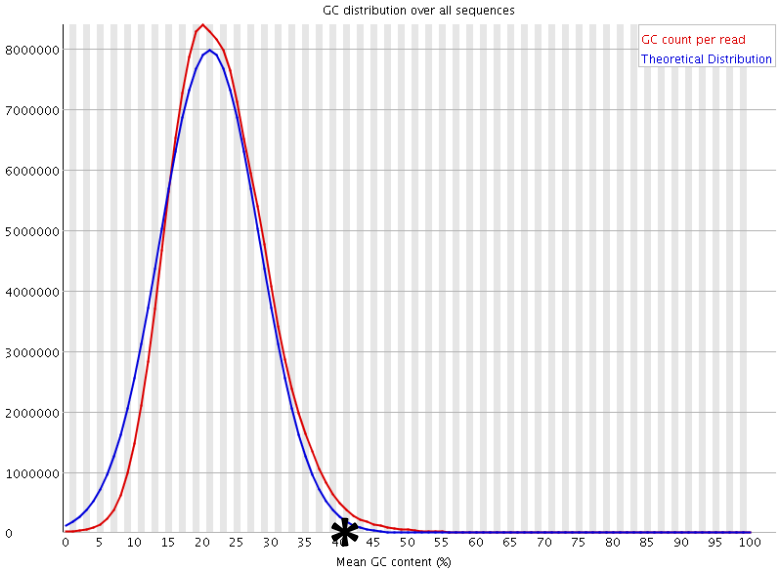
- x-axis = mean GC content (%)
- y-axis = # of reads
- red: observed read count, blue: theoretical distribution (given observed)

GOOD LANE

mouse genome ≈ 40% GC

BAD LANE





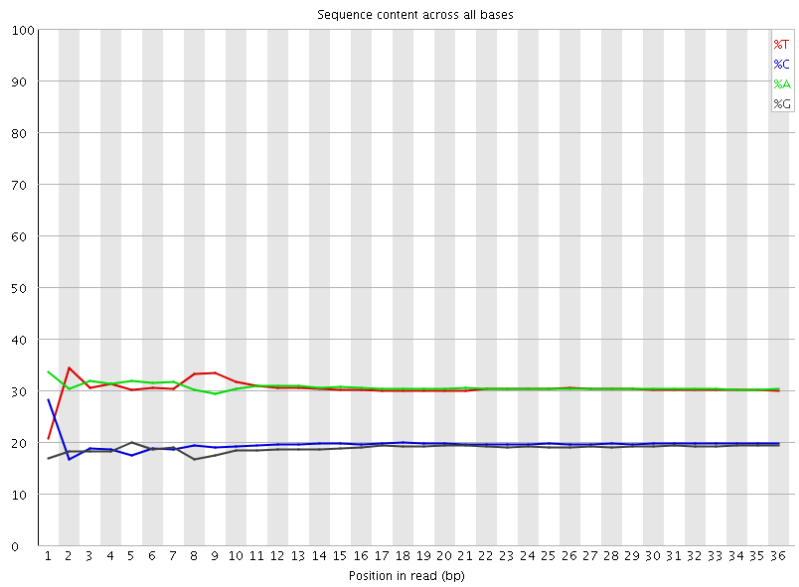
Can this be fixed? **No.**

NGS QC: Sequence bias across read length. (1)

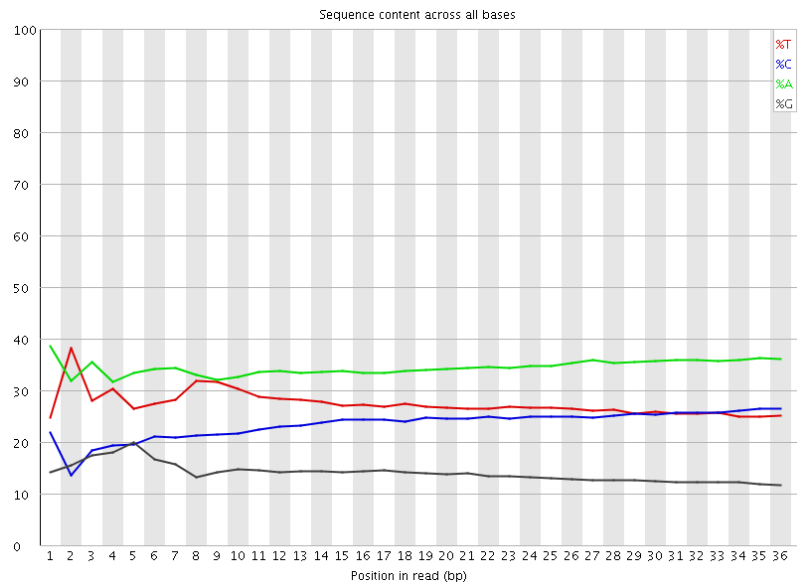
This plot shows the nucleotide distribution per read position for all reads in a lane.

- x-axis = position in read (bp)
- y-axis = % of all reads in the lane
- colors refer to individual nucleotides: **A**, **C**, **G**, **T**

GOOD LANE



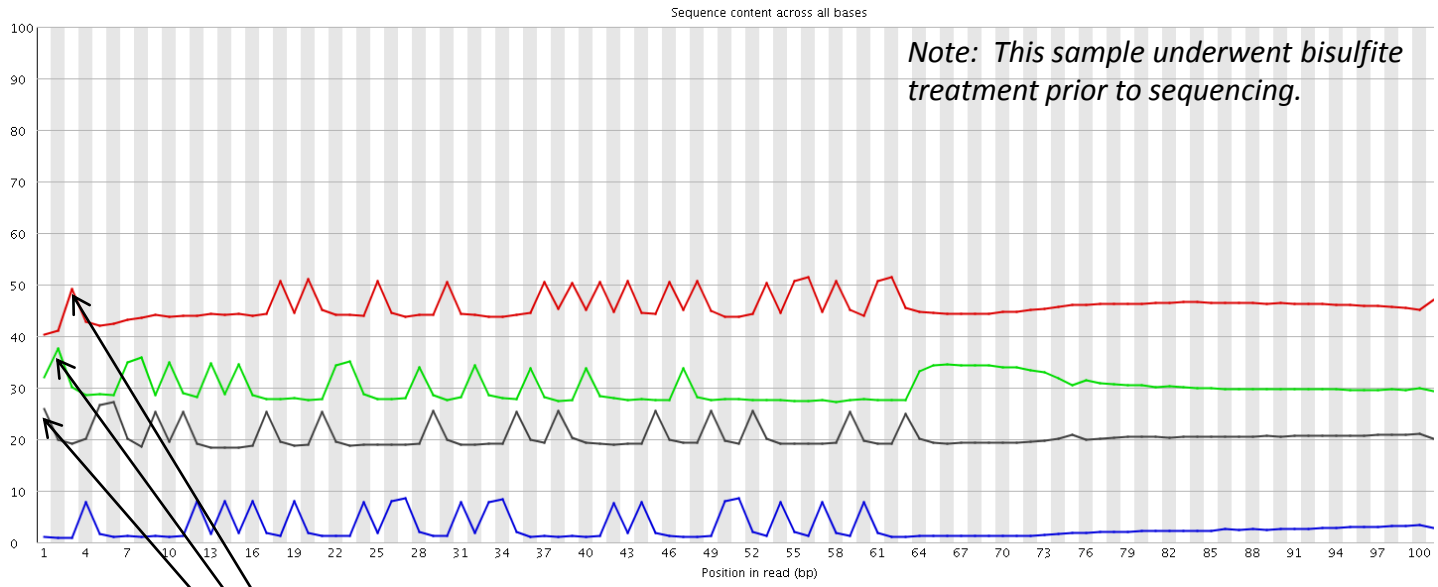
BAD LANE



Can this be fixed? **No.**

NGS QC: Sequence bias across read length. (2)

This lane has a different problem – one sequence motif is highly over-represented.



primer/adapter sequence: **GATCGGAAGAGCACACGTCTGAACTCCAGTCACACAGTGATCTCGTATGCCGTCTTCTGCTTG**

In this lane, ~10% of reads have the adapter sequence & the rest are normal.

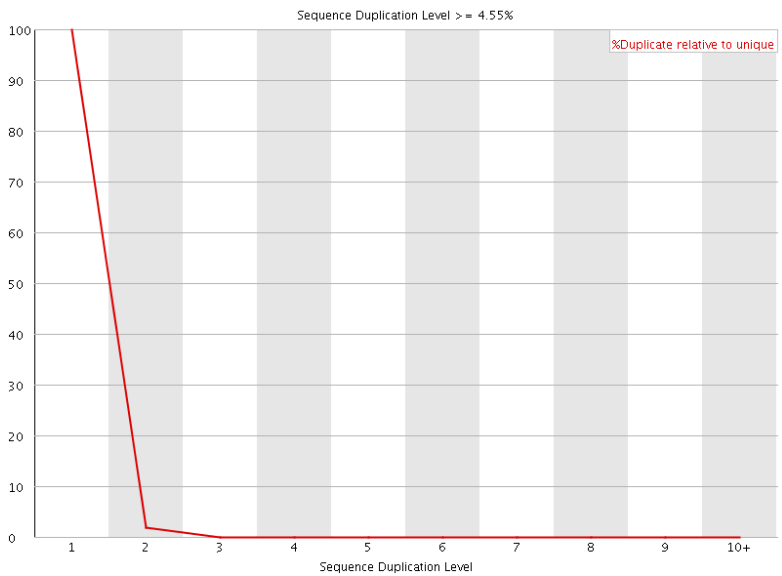
<u>Can this be fixed?</u> Yes. Simply remove the reads w/adapter contamination, and everything that's left should be fine. (Talk to a bioinformatics analyst for help.)

NGS QC: Low uniqueness among reported reads.

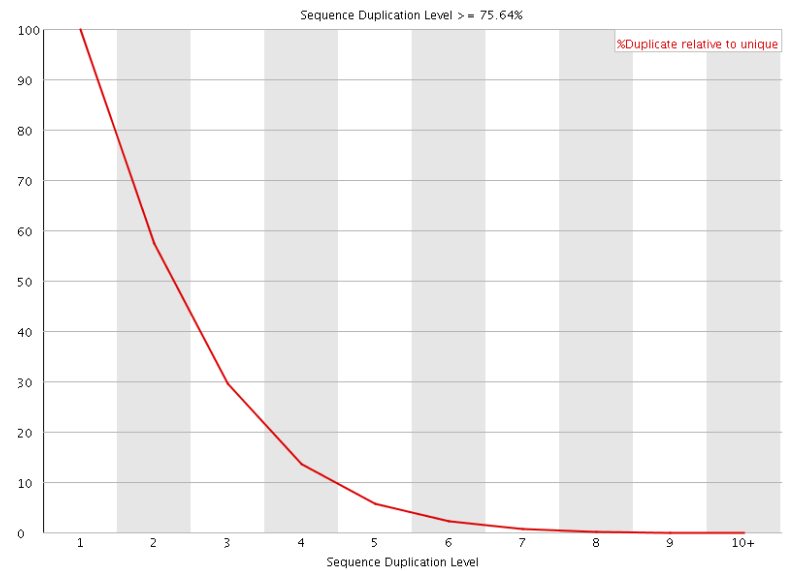
This plot shows the degree of duplication for a subset of reads in a lane.

- x-axis = sequence duplication level
- y-axis = % duplicates relative to unique reads

GOOD LANE



BAD LANE



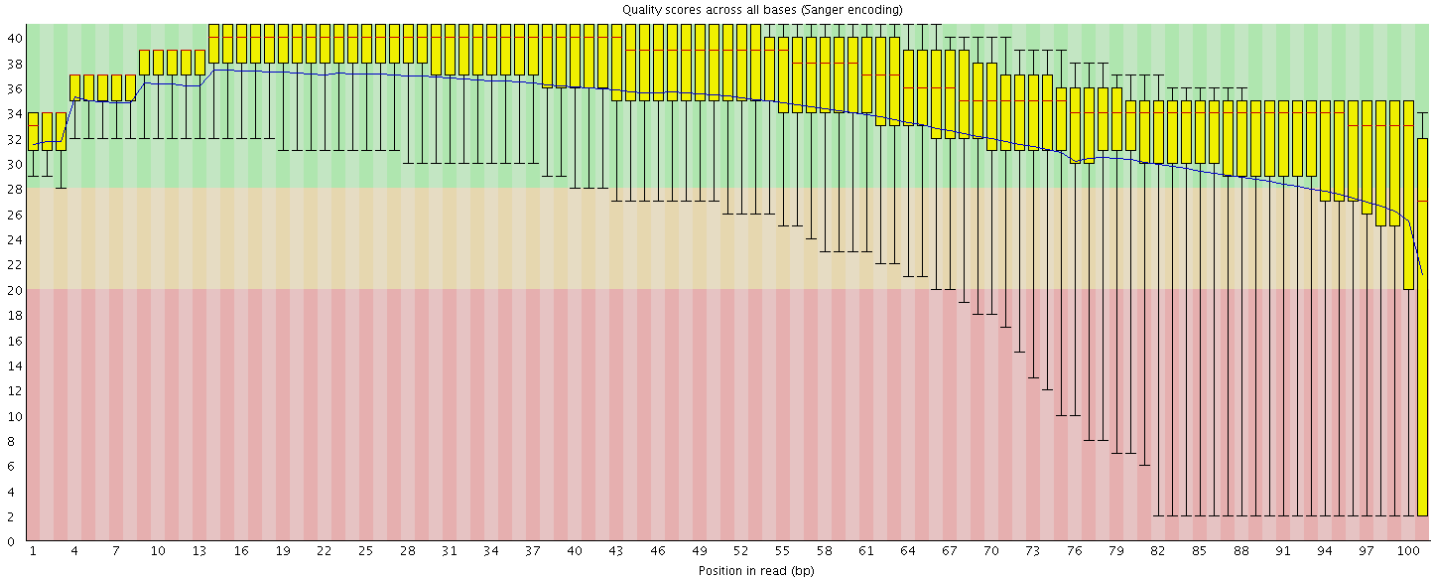
Can this be fixed? **Maybe.**

NGS QC: Base quality score per read position. (1)

This plot shows the base quality score distribution for all reads in a lane, with each read position considered independently.

- x-axis = position in read (bp)
- y-axis = Phred-like base quality score [pink=0-20, tan=20-30, green=30-40]
- red bar = median score, blue line = mean score
- yellow box = 25th to 75th percentile, black whiskers = 10th to 90th percentile

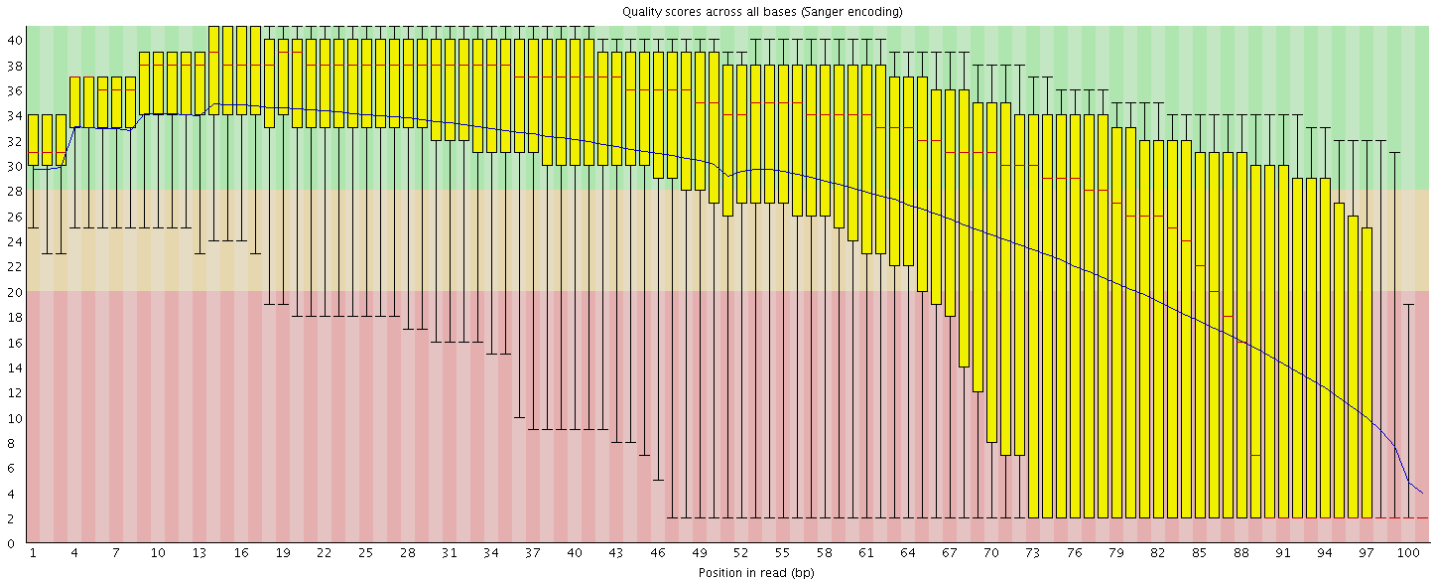
GOOD/NORMAL LANE



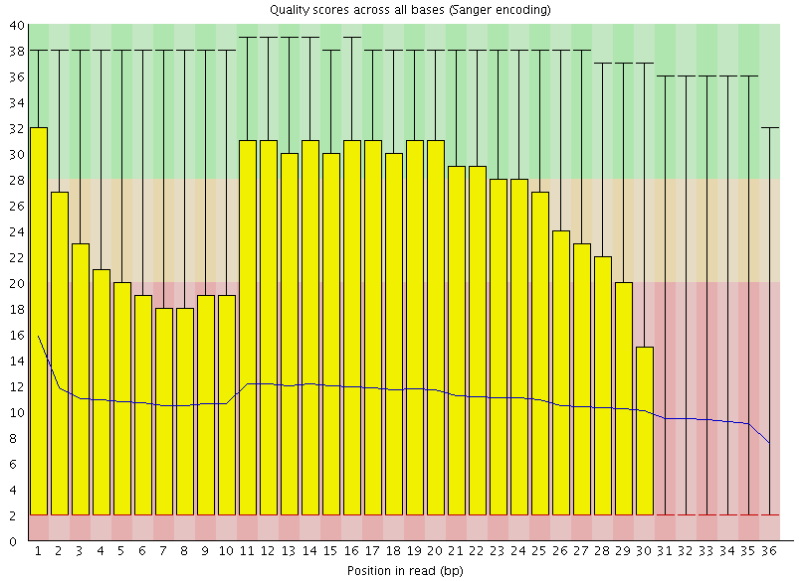
NGS QC: Base quality score per read position. (2)

SALVAGEABLE LANE

[trim very bad positions from the 3' end]



FAILED LANE



Quality Trimming & Filtering Example [Step 1]

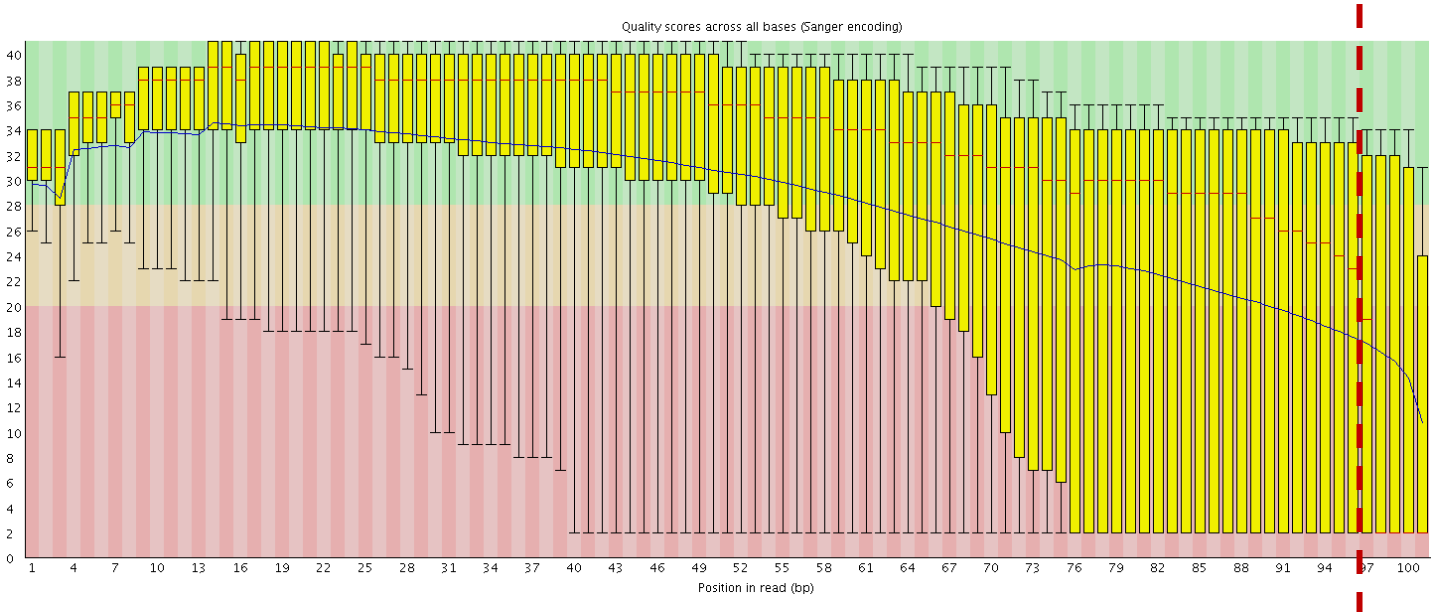
Most lanes will not have problems with sequence bias, GC content, adapters, etc.

Most lanes <u>will</u> have reads with base quality problems. Here is a typical example…

Note: Stringency of base qualities to retain is somewhat application-specific. Talk to a bioinformatics analyst for help with your particular experiment/analysis.

Step 1 = Trimming by base quality.

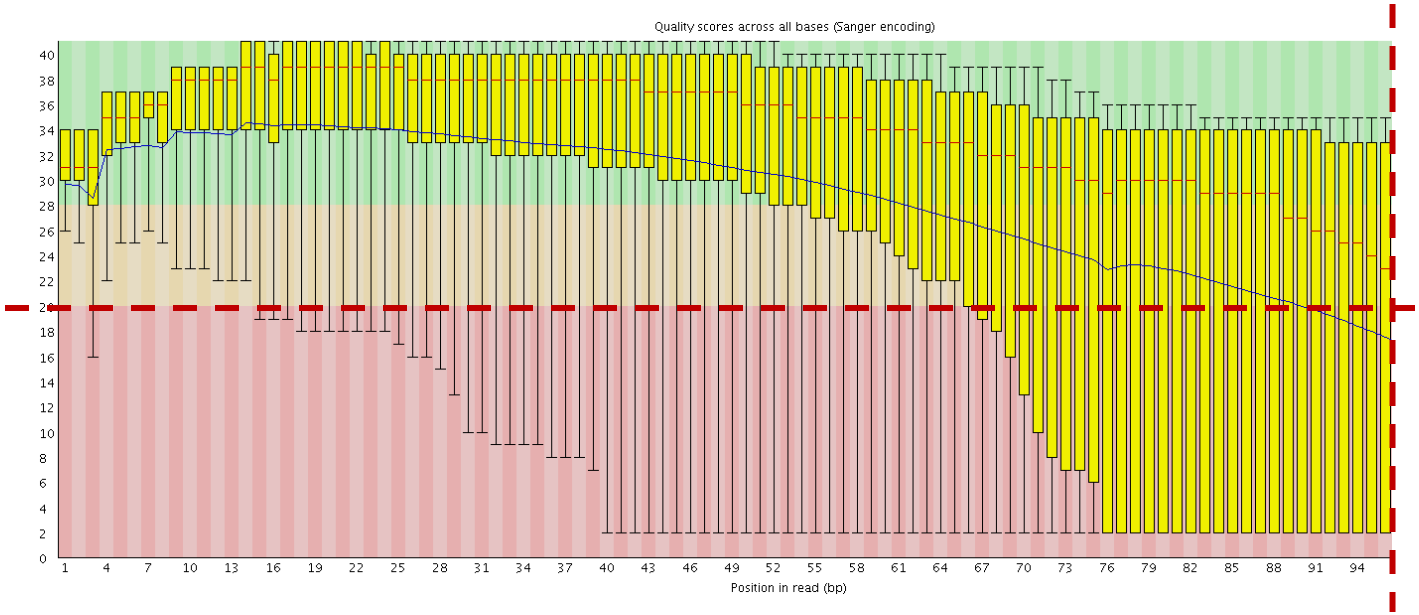
My rule of thumb: Trim reads where the median base quality falls below 20.



Quality Trimming & Filtering Example [Step 2]

Step 2 = Filtering by base quality.

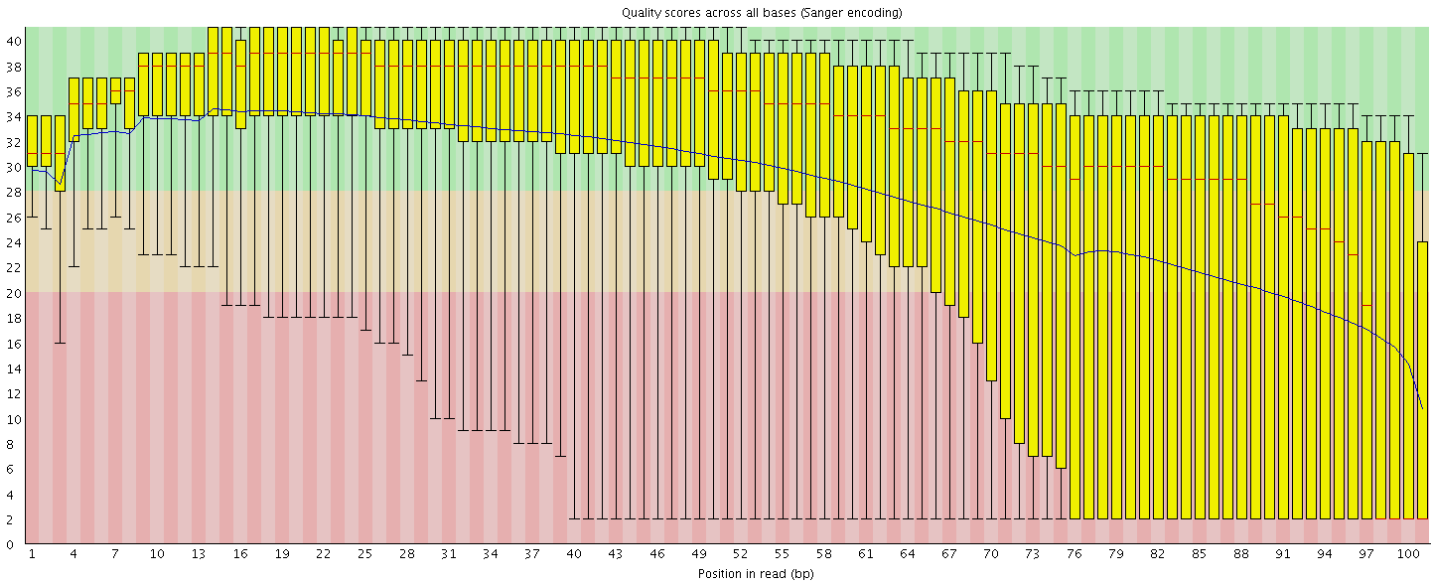
My rule of thumb: Retain only reads with an average base quality score ≥ 20.



Quality Trimming & Filtering Example [Result]

BEFORE:

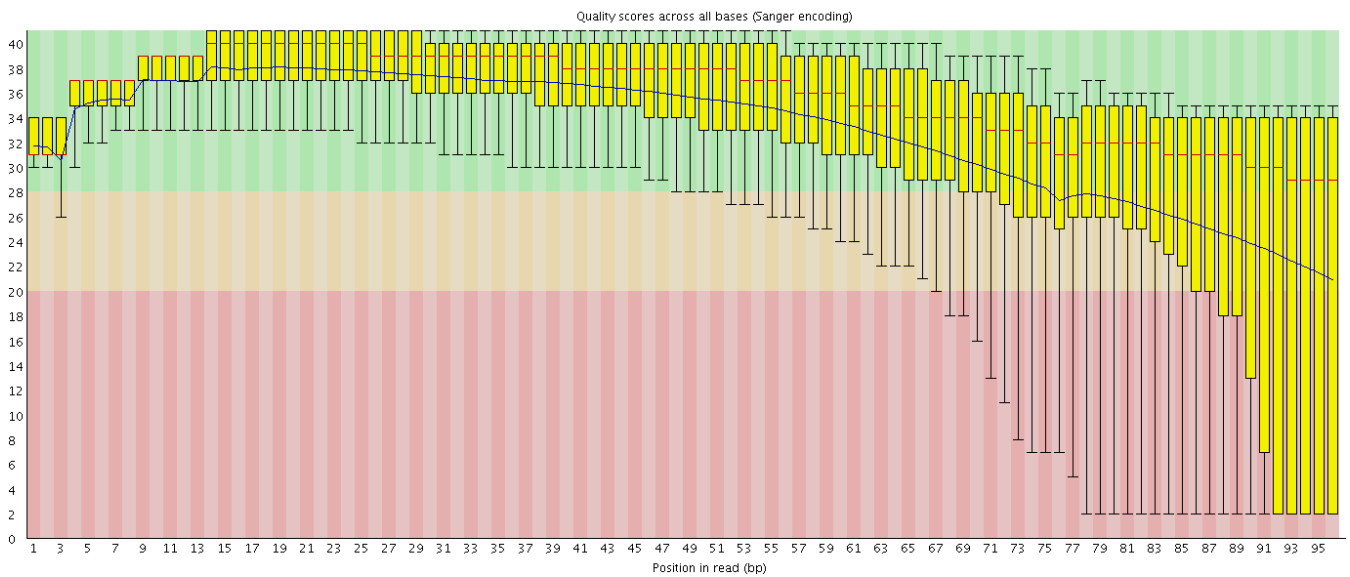
Raw data (i.e. untrimmed & unfiltered) →



AFTER:

Post-trimming & post-filtering base quality distribution →

~82% of reads in this lane pass this QC filter.



Quality Trimming & Filtering: Why Bother?

"This seems like a lot of extra work. Is it really necessary?"

Why trim? Most NGS aligners perform global, not local, alignments – which means it attempts to match the reference from base 1 to base N of the read. If the # of base errors exceeds the allowed mismatches, the alignment will be discarded. ***Trimming likely erroneous bases from the end of reads may salvage alignment.***

Why filter? If there are many erroneous base calls in a read, it could be mapped to the wrong place. More likely, it won't be aligned at all. ***By filtering, you reduce the input set to reads you reasonably expect will align to a reference genome… so you can more easily determine whether your overall mapping efficiency is within normal ranges.***

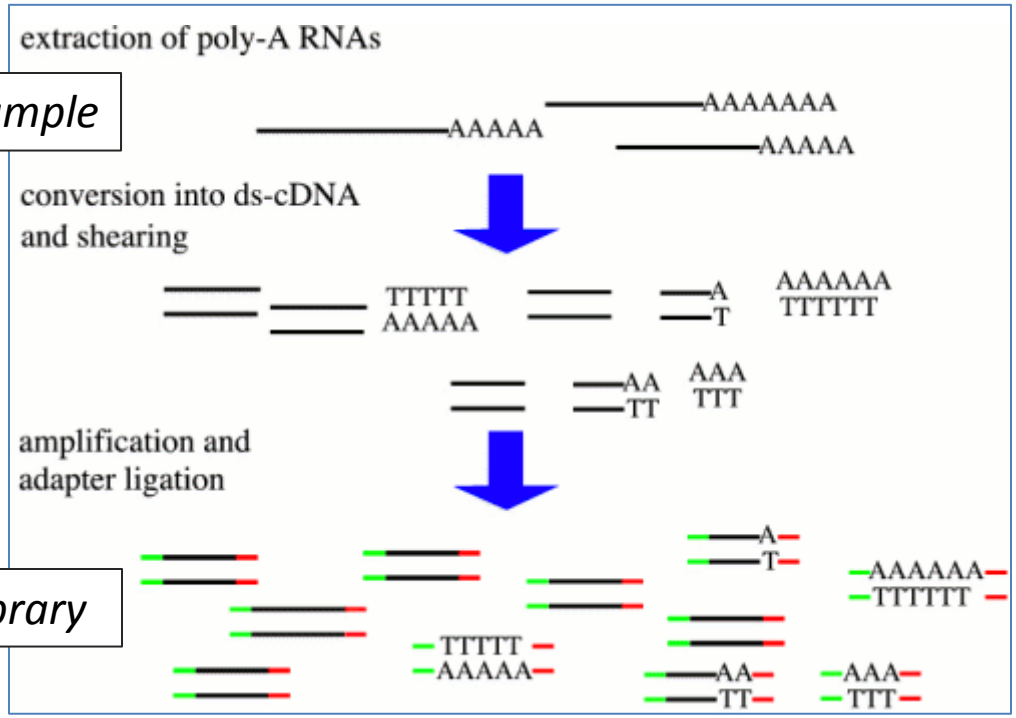
How long will this take? Depending on the size of your raw sequencing files, maybe as long as a few hours. (This is practically nothing relative to how long the downstream mapping & analysis tasks will take…)

Topic 3
RNA-seq

General Assumption of RNAseq…

The relative abundance of what's in here…

Biological Sample



…is the same as what's in here.

Sequence Library

http://cmb.molgen.mpg.de/2ndGenerationSequencing/Solas/RNA-seq.html

What RNAseq can tell you.

Standard analysis:

- genome-wide transcript abundance
- differences between individuals, tissues, or treatment conditions
  - differential abundance
  - differential isoform usage
- discovery of novel isoforms

Other analysis:

- de novo transcriptome assembly
- allele-specific expression // SNP identification

RNAseq vs Microarray

Advantages over microarray:

- not limited to what you already know (novel isoforms, novel genes, etc)
- bigger dynamic range
- can distinguish abundance of different isoforms at the same locus

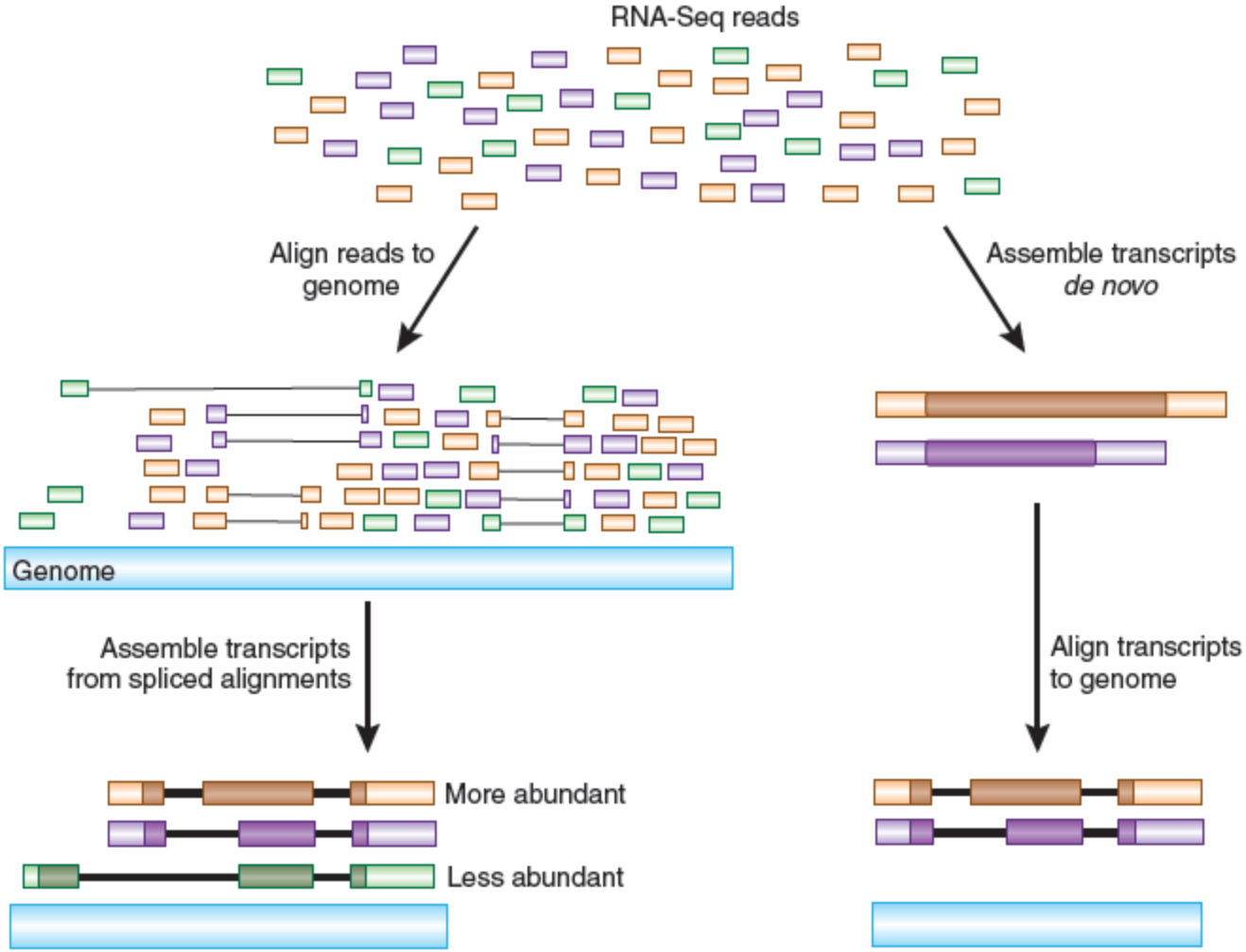
Kinds of RNAseq

There are a number of variations of the "RNAseq" experiment:

- mRNA-seq [polyA enriched protocol]
- total RNA-seq
- strand-specific RNA-seq
- small RNA-seq

- This is what is typically done at NISC.
- This is the focus of the rest of this module.

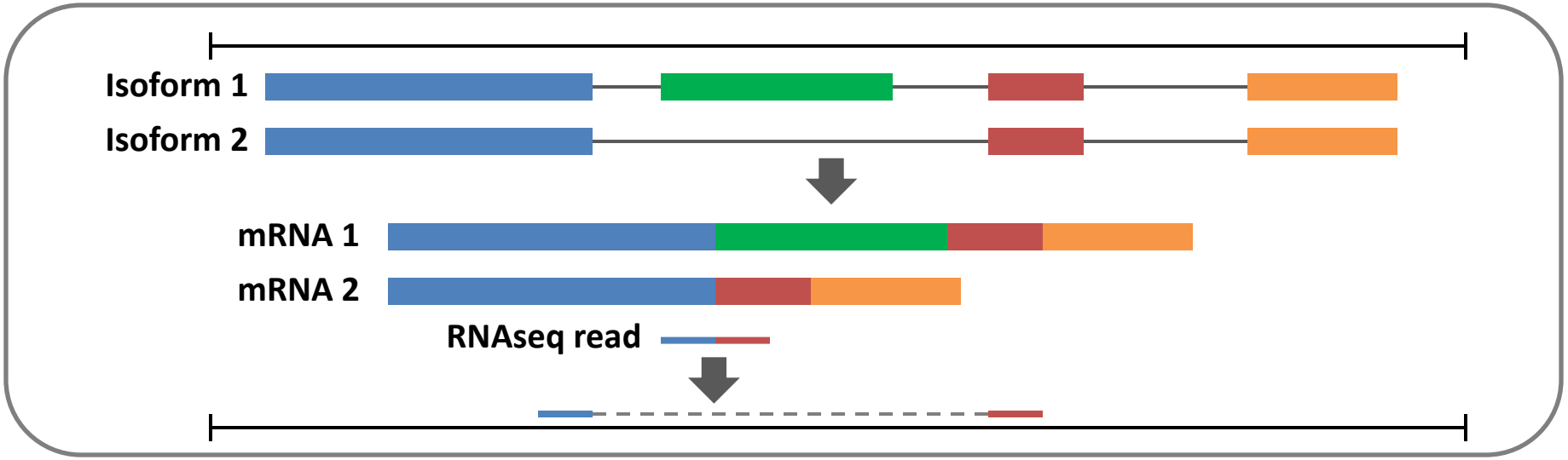
Two general RNAseq analysis approaches.



Nat. Biotech 28:421-423.

RNAseq Analysis, Part 1: Mapping to Reference Genome

Primary Challenge: Mapping across exon juctions.



- Some mapping tools don't allow any gaps, not even a single-base indel.
- Some mapping tools can place small indels, but the accuracy of these alignments tends to be poor… especially with indels longer than a few bases.
- Mapping tools cannot accurately identify large gaps (i.e. introns) in an efficient manner. [At least not yet.]

Why? The reads are too short to confidently anchor both sides of an alignment… unless the aligner knows "where to look".

Mapping RNAseq reads to a reference genome: TopHat.

TopHat will align reads to a reference genome, including mapping across an exon junction. A reference annotation of known genes/transcripts is not required.

1) Reads that map contiguously to the gene are identified first. These hits are used to define potential exons.

identify candidate exons via genomic mapping

2) A database of possible splice junctions is built based on those potential exons.

Generate possible pairings of exons

3) Mapping of the remaining reads against those junctions is attempted.

Align "unmappable" reads to possible junctions

Trapnell et al. 2008. Bioinformatics 25:1105-1111.
http://tophat.cbcb.umd.edu/

TopHat Parameters

Required:

<ref_index>	**reference genome indexed for bowtie/bowtie2**
<fastq1>	**mate1 reads in FASTQ format**
<fastq2>	**mate2 reads in FASTQ format (in same order as mate1 reads)**
--mate-inner-dist	**expected (mean) inner distance between mate pairs**

Bowtie-related options:
--bowtie1
--b2-very-fast
--b2-
--b2-
--b2-
--b2-
--b2-

Library descriptors:
--solexa-quals
--solexa1.3-quals
--phred64-quals
--integer-quals

User-defined transcriptome options:
-G/--GTF
--transcriptome-index
-j/
--n
-T/
-x/
-M/

Mapping search/stringency options:
--bowtie-n
--segment-length
--genome-read-mismatches
--transcriptome-mismatches
ad-mismatches
gment-mismatches
low-partial-mapping
x-multihits
-coverage-search
verage-search
cro
-mu
-ig
-do

Junction search options:
--min-anchor
--splice-mismatches
--min-intron-length
--max-intron-length
--min-coverage-intron
--max-coverage-intron
--min-segment-intron
--max-segment-intron

Indel control options:
--max-insertion-length
--max-deletion-length
ertions
etions
novel-indels

Output options:
--output-dir
--report-secondary-alignments
--report-discordant-pair-alignments
--no-sort-bam
--no-convert-bam
--keep-fasta-order
--rg-id
--rg-sample
--rg-library
--rg-description
--rg-platform-unit
--rg-center
--rg-date
--rg-platform

Computational/Processing options:
--num-threads
--keep-tmp
--tmp-dir
--zpacker
--unmapped-fifo

Talk to a bioinformatics analyst for help…

Example RNAseq Analysis: Prep raw data files.

This example includes 4 PE 101mer lanes from the HiSeq. There are two replicates each of male & female samples. All data is mouse liver tissue.

A) Run FastQC to check sequence bias, GC content, etc.

(Let's assume everything is fine here for these lanes.)

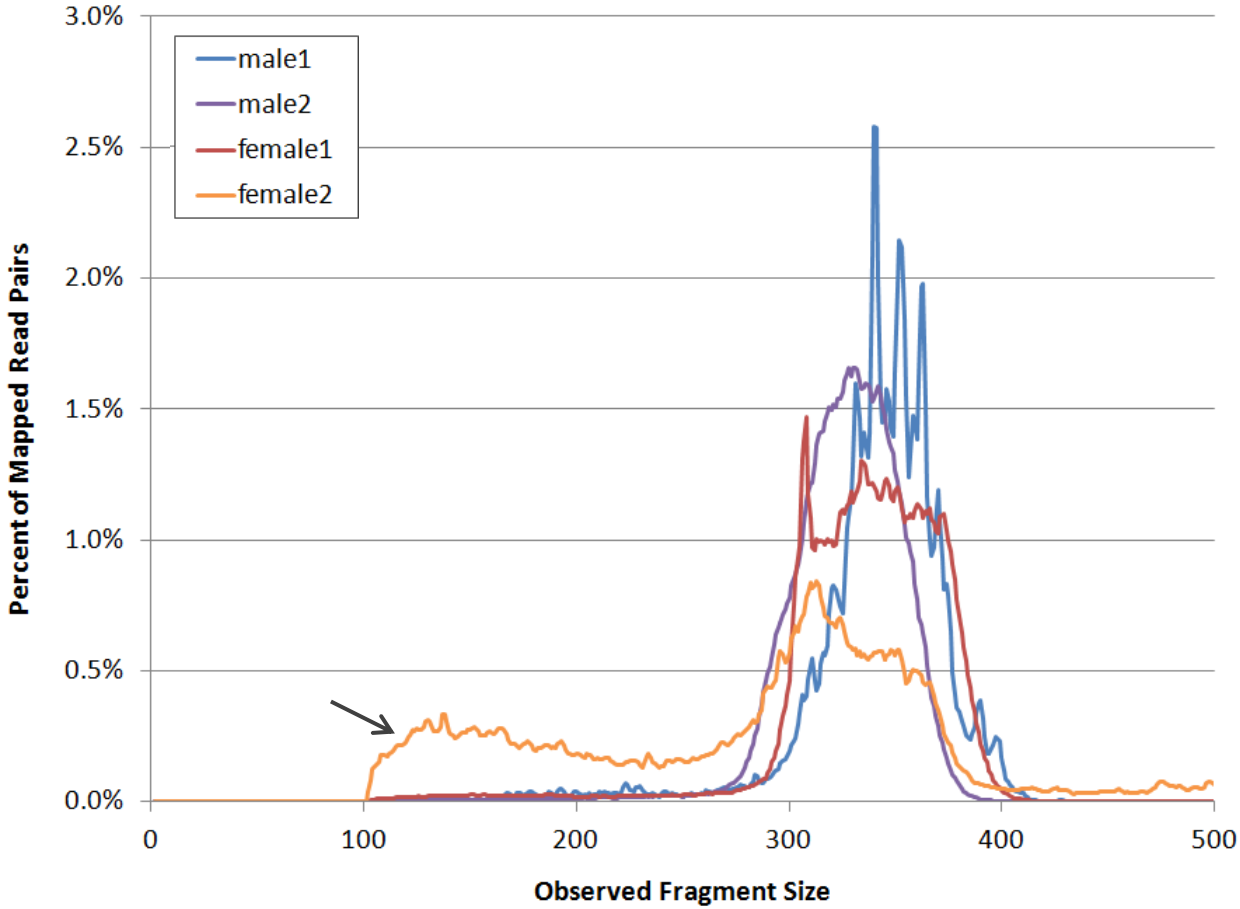
B) Trim & filter raw reads. Since these reads are paired, require both mates to pass the average base quality filter @ Q20.

Sample	Retained Positions in Trimmed Mate1	Retained Positions in Trimmed Mate2	All Read Pairs	Read Pairs Passing Filter	% Passing Filter
male1	1-101	1-101	201,116,634	**154,557,046**	76.9
male2	1-101	1-101	190,176,759	**168,186,082**	88.4
female1	1-101	1-101	201,411,830	**157,440,703**	78.2
female2	1-101	1-101	220,917,100	**188,013,701**	85.1

Example RNAseq Analysis: Find fragment size per library.

C) Determine estimated fragment size. Options are…

* use expected size based on library preparation
* empirically determine: map a subset of reads to a known transcriptome



Example RNAseq Analysis: Run TopHat. (1)

D) Select appropriate options/parameters & launch TopHat.

Then wait… and wait… and wait. One lane of HiSeq data will take 2-7 days, depending on selected options & server availability.

Output files:

accepted_hits.bam

unmapped.bam

deletions.bed

insertions.bed

junctions.bed

logs/

prep_reads.info

SAM [Sequence Alignment/Map]

This is the **<u>standard format for aligned NGS data</u>**.

BAM = binary version of the same format.

Sample	SAM size	BAM size
male1	116 G	23 G
male2	127 G	24 G
female1	109 G	23 G
female2	130 G	25 G

Example RNAseq Analysis: Run TopHat. (2)

D) Select appropriate options/parameters & launch TopHat.

Then wait… and wait… and wait. One lane of HiSeq data will take 2-7 days, depending on selected options & server availability.

Output files:

accepted_hits.bam

unmapped.bam

deletions.bed

insertions.bed

junctions.bed

logs/

prep_reads.info

The insertions, deletions, and junctions identified by TopHat, in standard BED format.
For example...

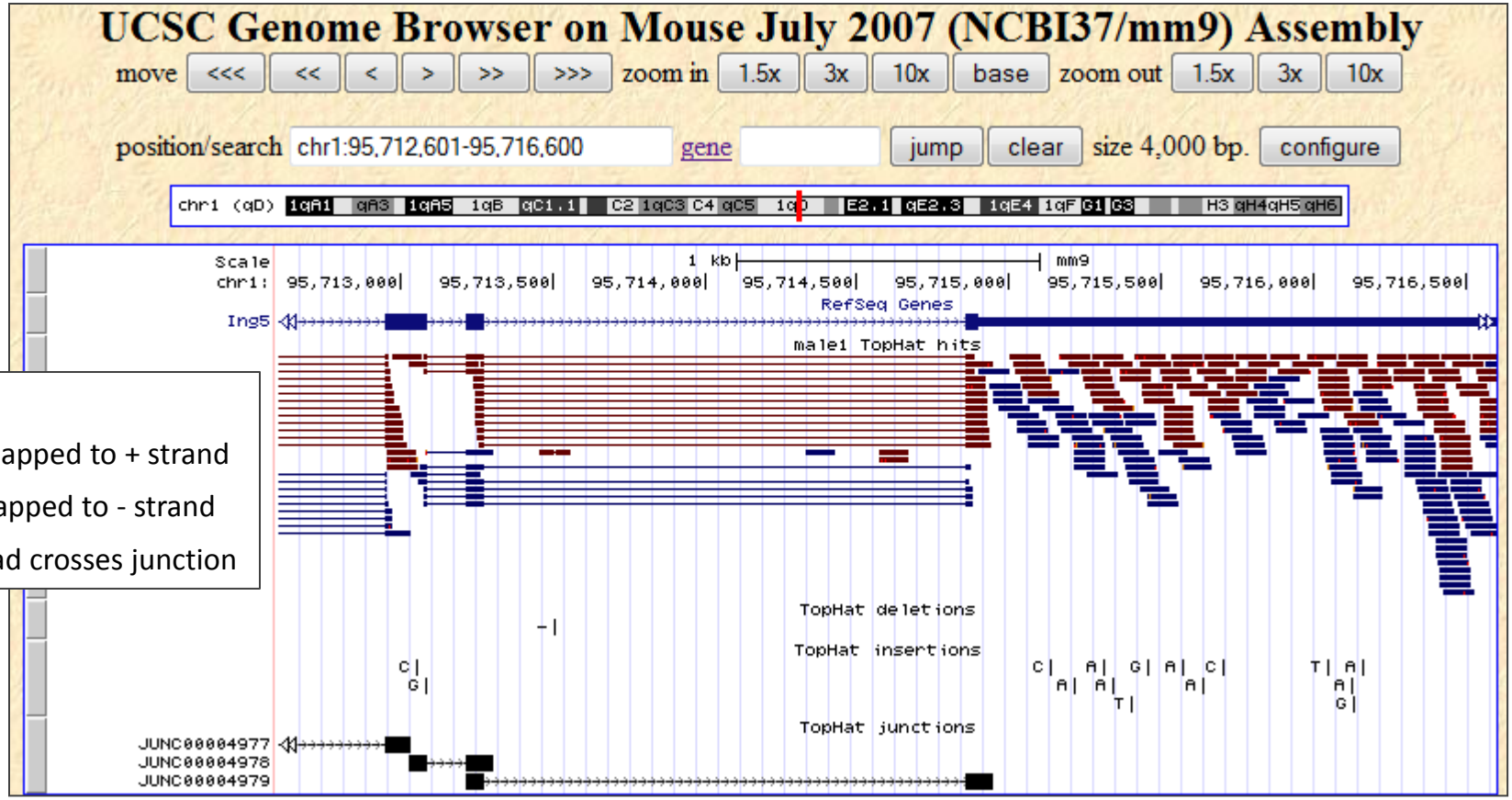
```
track name=insertions description="TopHat insertions"
chr1    3206248    3206248    T             1
chr1    4481575    4481575    G             1
chr1    4678425    4678425    C             2
chr1    4679317    4679317    CCCAGCGGCT    5
chr1    4679318    4679318    CCAGCGGCTC    3
chr1    4761209    4761209    A             3
```

Processing info & log files.
Most users will never look at these.

TopHat Output (Viewed in UCSC Genome Browser)

BED & BAM* formats can both be visualized in the UCSC Genome Browser.

Here is a screen shot for the “male1” sample:



KEY

Blue = read mapped to + strand

Red = read mapped to - strand

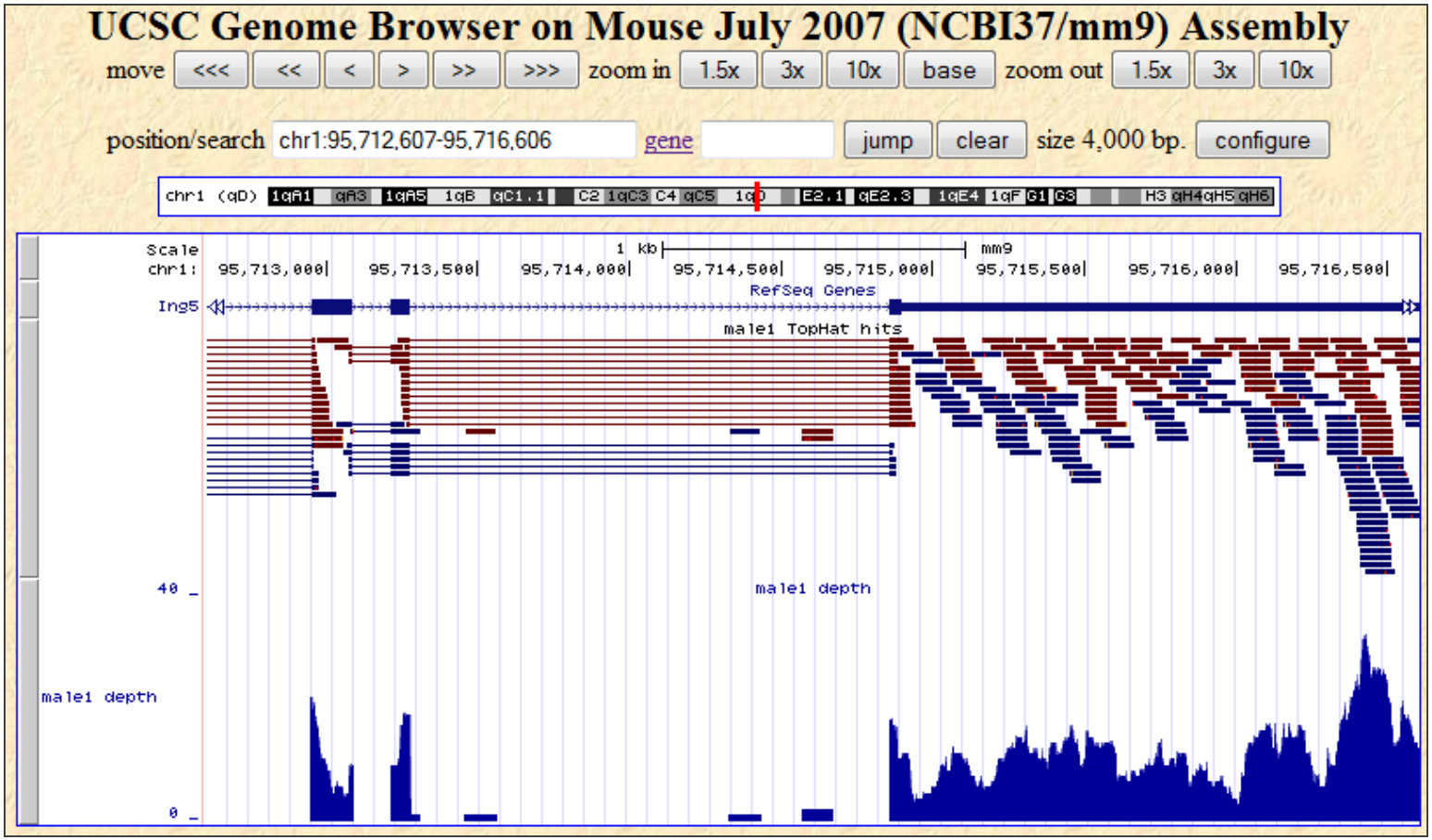
Thin line = read crosses junction

** BAM files are not uploaded directly, but are instead stored in a web-accessible server location. At NIEHS, getting this set up requires the help of the science IT group…*

Visualization of RNAseq Data [UCSC Genome Browser]

BAM view shows per-read mapping, but is limited to regions with fairly low coverage. Converting to aggregate depth is the more standard visualization.

[Aggregate coverage tracks can be generated with command-line tools from the BEDtools package & UCSC scripts.]



TopHat Output "Reality Check"

A quick summary look at the mappings per lane can confirm that the results are in the typical ranges for your particular sample type and organism.

Reads With At Least One Mapped Hit

Sample	Read Pairs Passing Base Quality Filter	Reads Passing Base Quality Filter	Reads With 1+ Hits	% of Reads With 1+ Hits
male1	154,557,046	309,114,092	253,058,581	71.6%
male2	168,186,082	336,372,164	285,502,351	75.4%
female1	157,440,703	314,881,406	281,728,441	77.9%
female2	188,013,701	376,027,402	326,951,462	82.2%

Total Mapped Hits [note: a given read can be mapped to multiple locations]

Sample	Read Pairs Passing Base Quality Filter	Reads Passing Base Quality Filter	Total Mapped Hits	Avg Hits Per Reads	Avg Depth Per Transcriptome Position*
male1	154,557,046	309,114,092	354,893,696	1.1	438x
male2	168,186,082	336,372,164	392,489,789	1.2	482x
female1	157,440,703	314,881,406	337,475,059	1.1	415x
female2	188,013,701	376,027,402	400,658,799	1.1	495x

**Estimated average depth assumes transcriptome is 3% of genome size.*

RNAseq Analysis, Part 2: Quantification.

Primary Challenge: Convert mapped fragments to a numerical score for abundance.

FPKM = Fragments Per Kilobase (of transcript) per Million mapped fragments

-- A method to normalize mapped read counts both by the size of the transcript (or other genomic feature) and by the total size of the given lane of data.

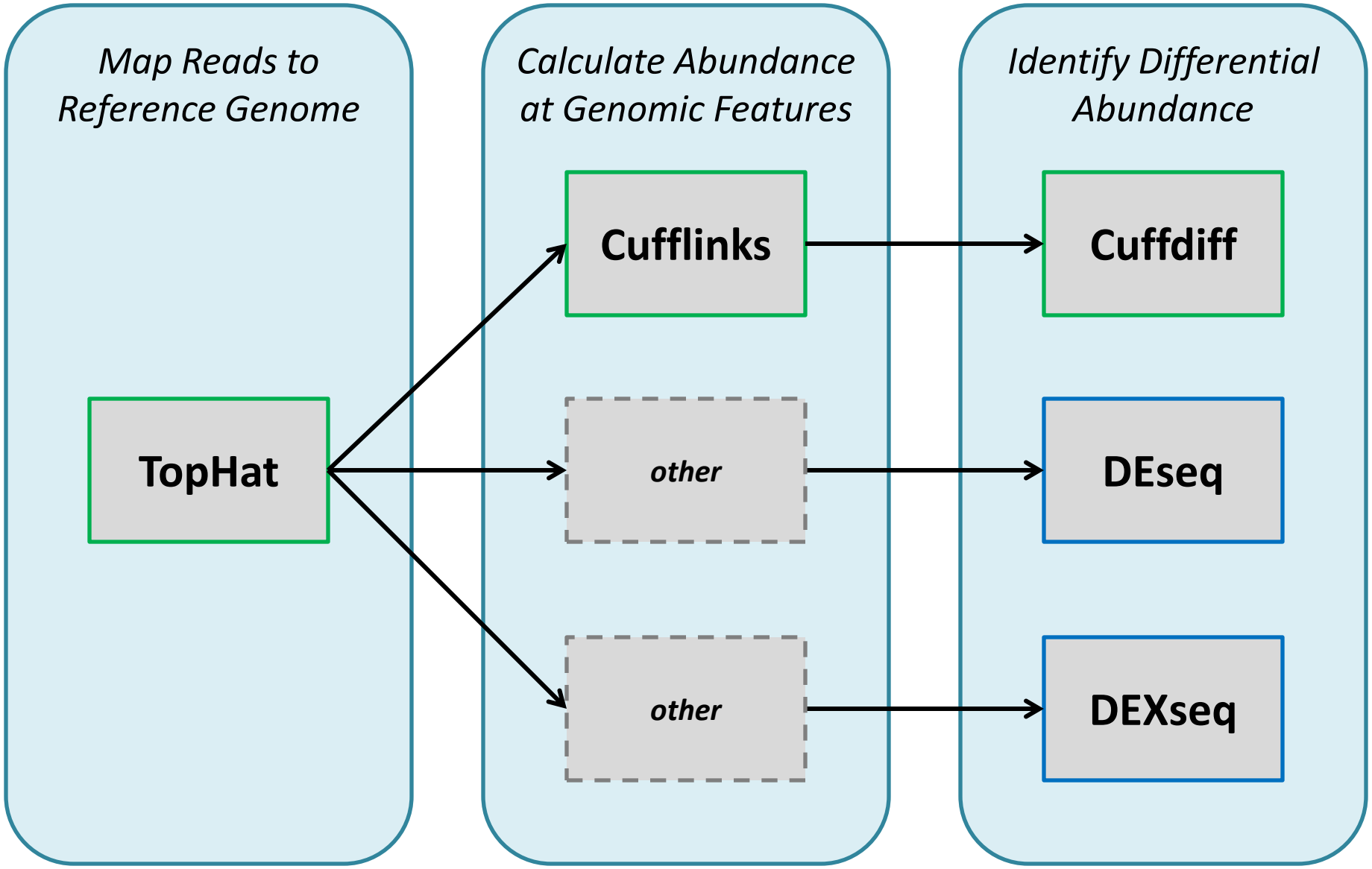
The same number of fragments are mapped to each transcript, but Transcript1 is twice the length of Transcript2…

So the FPKM for Transcript1 will be ½ the FPKM for Transcript2.

Most Popular RNAseq Quantification Tools

	Cufflinks	DEseq
Tools.	***Cufflinks***: Simultaneously evaluates abundance (as FPKM) at the gene & transcript levels. ***Cuffdiff***: Simultaneously evaluates differential genes, isoforms, promoters, splicing, & coding sequences.	***DEseq***: Identifies differentially expressed genomic features, <u>for use at gene level</u>. ***DEXseq***: Identifies differentially expressed genomic features, <u>for use at exon level</u>.
Input format.	No formatting changes necessary. (Part of same suite as TopHat/Bowtie.)	Requires significant pre-processing to alter input format.
Platform.	Linux/Unix command-line.	R package.
Can find novel genes/transcripts?	Yes. Defining reference annotations is optional.	No. Must define reference annotations.
Output.	Can be difficult to interpret & parse.	Clean, complete, & user-friendly.
Relative speed.	Slower.	Faster.

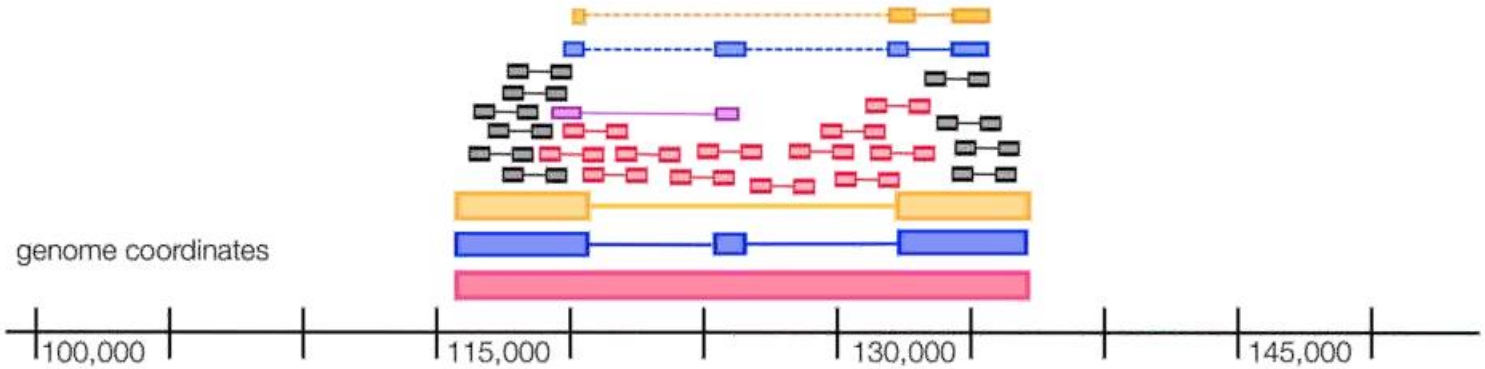
Quantification Workflow Overview



Cufflinks

Assigning reads to isoforms

Problem: infer which transcript each fragment came from



Some fragments could have come from any transcript (black), while others only one (blue, yellow). The purple fragment could have come from either the red or the blue one.

If you're interested in learning about the statistics behind how these abundance estimations are made, check out these references…

- Trapnell et al. 2010. Nat Biotech 28(5):511-515.
- Roberts et al. 2011. Bioinformatics 27(17):2325-2329.

Cufflinks Parameters

Required:
<~.sam/~.bam> **aligned reads in SAM or BAM format**

General Options:
-o/--output-dir
-p/--num-threads
--library-type
--seed
-b/--frag-bias-correct
-u/--multi-read-correct
-v/--verbose
-q/--quiet
--no-update-check

Reference-Guided Assembly Options:
-G/--GTF
-g/--GTF-guide
-M/--mask-file
--no-faux-reads
--3-overhang-tolerance
--intron-overhang-tolerance

Abundance Estimation Options:
-m/--frag-len-mean
-s/--frag-len-std-dev
-N/--upper-quartile-norm
--max-mle-iterations
--num-importance-samples
--compatible-hits-norm
--total-hits-norm

Assembly Options:
-L/--label
-F/--min-isoform-fraction
-j/--pre-mrna-fraction
-I/--max-intron-length
-a/--junc-alpha
-A/--small-anchor-fraction
--min-frags-per-transfrag
--overhang-tolerance
--max-bundle-length
--max-bundle-frags
--min-intron-length
--trim-3-avgcov-thresh
--trim-3-dropoff-frac
--max-multiread-fraction
--overlap-radius

Example RNAseq Analysis: Run Cufflinks.

A) Select appropriate options/parameters & launch Cufflinks.

Then wait. One lane of mapped HiSeq data will take several hours (if reference-guided) or a few days (if not referenced-guided).

Output files:

genes.fpkm_tracking

isoforms.fpkm_tracking

transcripts.gtf

These files include the FPKM calculated for each gene or each transcript/isoform that is either included in the reference annotations (if provided) or was identified by cufflinks (if no reference annotation is provided).

Genomic locations of all evaluated transcripts in a standard format (GTF).

If reference annotations were provided, these entries should match the input file.

If no reference annotations were provided, each entry has some mapped-read support according to the cufflinks algorithm.

Reference-Guided Cufflinks Quantification (1)

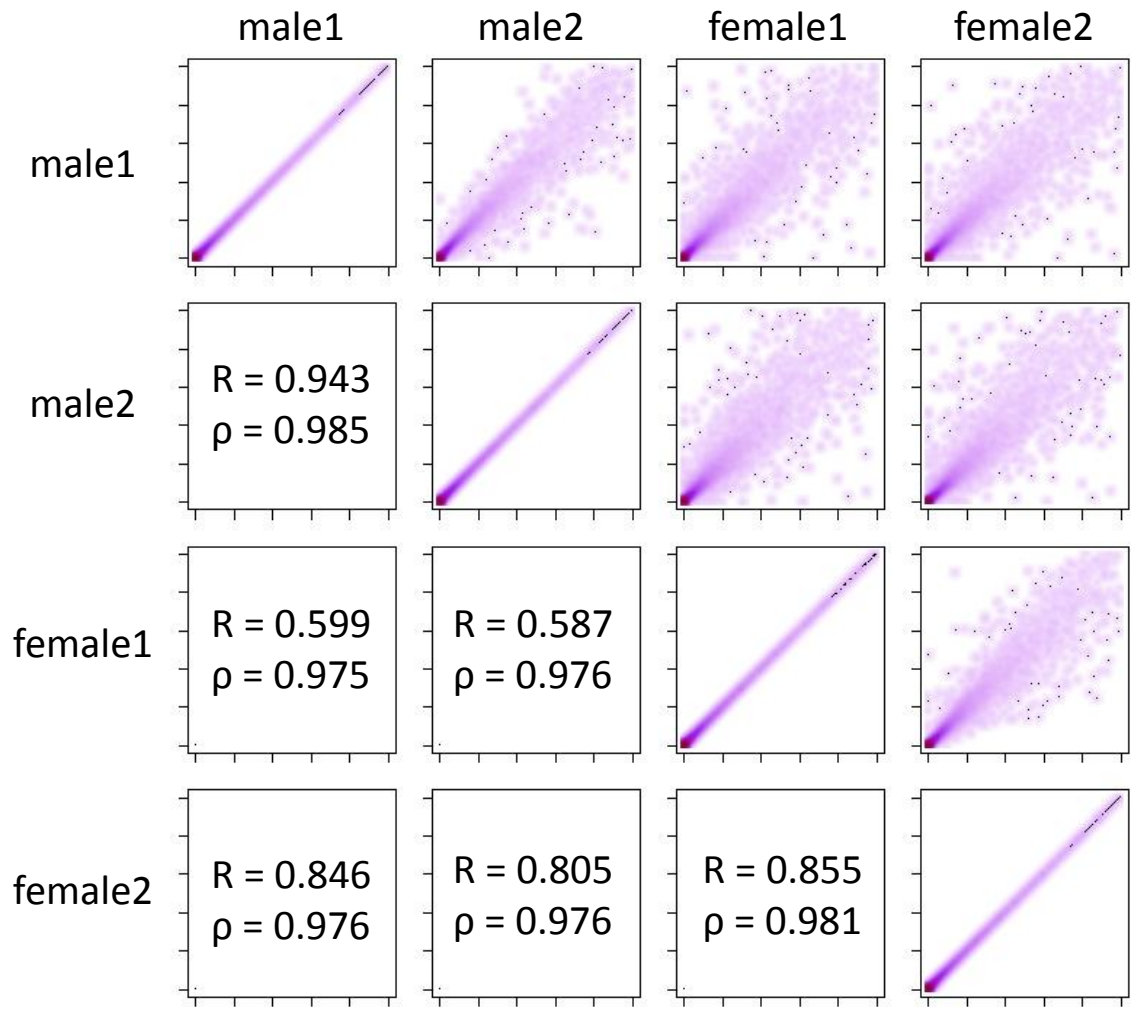
Comparison between lanes is straightforward because each sample is evaluated over exactly the same genomic features.

Example query: What is overall correlation between sample pairs?

The x- and y-axis are FPKM (0 to 100).

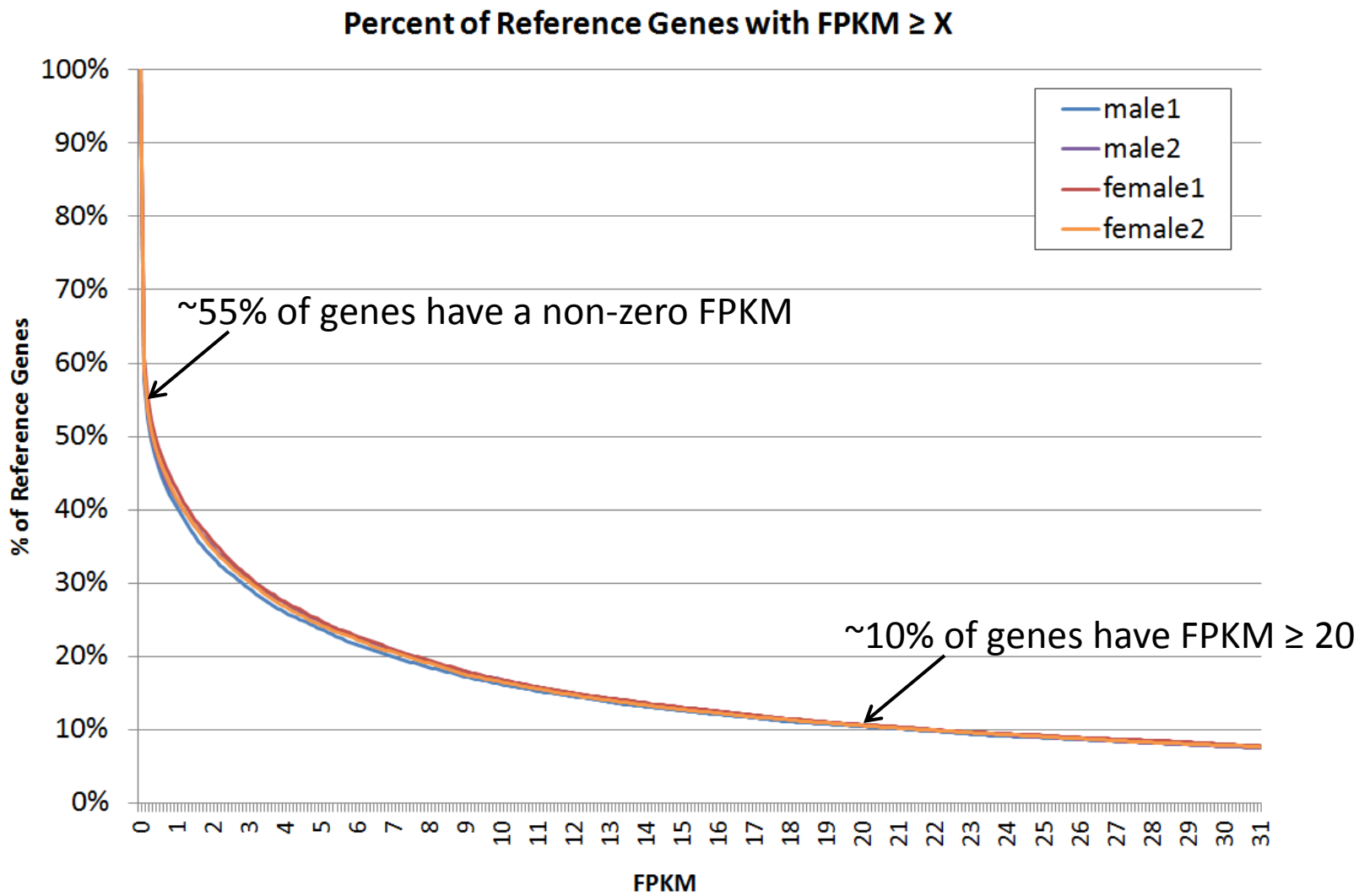
Each point is one gene. N=22,370 genes

Color intensity in the plot indicates density of points. *(e.g. bright spot in lower left corner = many genes with FPKM near zero for both samples)*



Reference-Guided Cufflinks Quantification (2)

Example query: What is distribution of calculated abundance (FPKM) per gene?



Cuffdiff Parameters

```
Required:
<transcripts.gtf>   genomic features to evaluate, must be in GTF format
<~.sam/~.bam>       aligned reads in SAM/BAM format [one per sample to compare]
```

```
General syntax:
cuffdiff [options] <transcripts.gtf> {list of SAM/BAMs}
```

With a slight adjustment to the input syntax, you can…

(a) consider all samples to be independent

```
{list of SAM/BAM files} =
male1.bam   male2.bam   female1.bam   female2.bam
```

(b) group replicate samples to be considered together

```
{list of SAM/BAM files} =
male1.bam,male2.bam   female1.bam,female2.bam
```

```
General Options:
-o/--output-dir
-T/--time-series
-c/--min-alignment-count
--FDR
--library-type
-M/--mask-file
-b/--frag-bias-correct
-u/--multi-read-correct
-N/--upper-quartile-norm
-L/--labels
-p/--num-threads
--seed
-m/--frag-len-mean
-s/--frag-len-std-dev
--num-importance-samples
--num-bootstrap-samples
--bootstrap-fraction
--max-mle-iterations
--compatible-hits-norm
--total-hits-norm
--poisson-dispersion
--no-update-check
--emit-count-tables
--max-bundle-frags
-v/--verbose
-q/--quiet
```

Example RNAseq Analysis: Run Cuffdiff.

A) Select appropriate options/parameters & launch Cuffdiff.

Then wait. Processing time depends on how many samples you are comparing; in general, run time is on the order of several hours.

Output files:

genes.fpkm_tracking
isoforms.fpkm_tracking
tss_groups.fpkm_tracking
cds.fpkm_tracking

Similar to the Cufflinks output, these files include the FPKM calculated for each entry of a given feature type (gene, isoform, etc).

cds.diff
cds_exp.diff
gene_exp.diff
isoform_exp.diff
promoters.diff
splicing.diff
tss_group_exp.diff

These files include evaluations (including significance tests) for differential abundance of each entry of a given feature type.

Identifying Differential Abundance from Cuffdiff Output

Example query: What genes are differentially expressed in males & females?

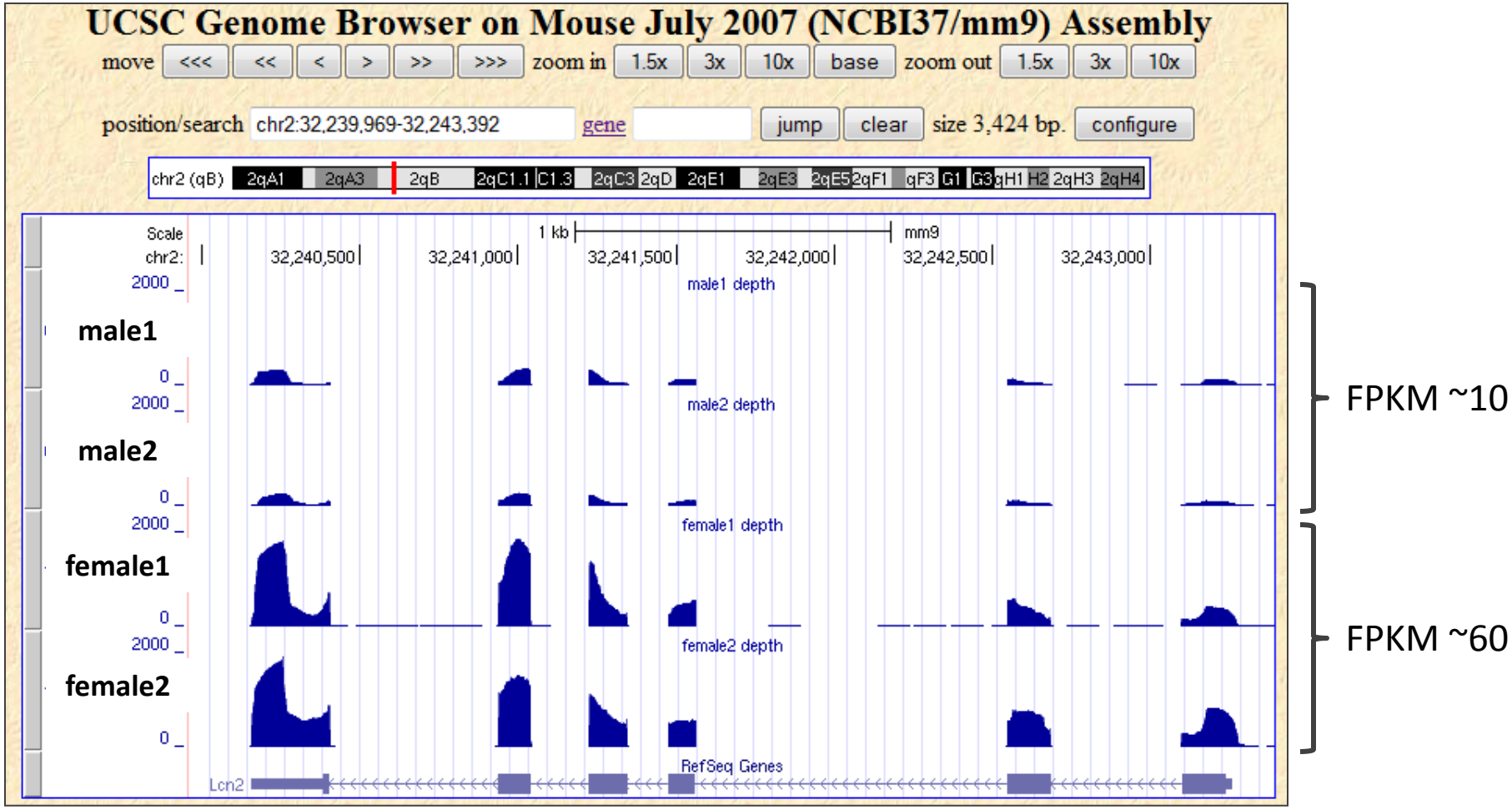
The genes_exp.diff file can be loaded in Excel. Here are the first few rows…

Gene	Locus	Sample1	Sample2	Status	FPKM 1	FPKM 2	log2 (fold change)	p-value	q-value	Significant
Lypla1|18777	chr1:4797973-4836816	q1	q2	OK	128.663	109.135	-0.23748	0.39102	0.65021	no
Tcea1|21399	chr1:4847774-4887990	q1	q2	OK	32.4786	29.7907	-0.12463	0.43895	0.68807	no
Atp6v1h|108664	chr1:5073253-5152630	q1	q2	OK	39.8703	41.4543	0.05621	0.82256	0.92118	no
Oprk1|18387	chr1:5578573-5596214	q1	q2	NOTEST	0	0	0	1	1	no
Rb1cc1|12421	chr1:6204742-6266185	q1	q2	OK	12.0806	14.6324	0.276475	0.27665	0.54847	no

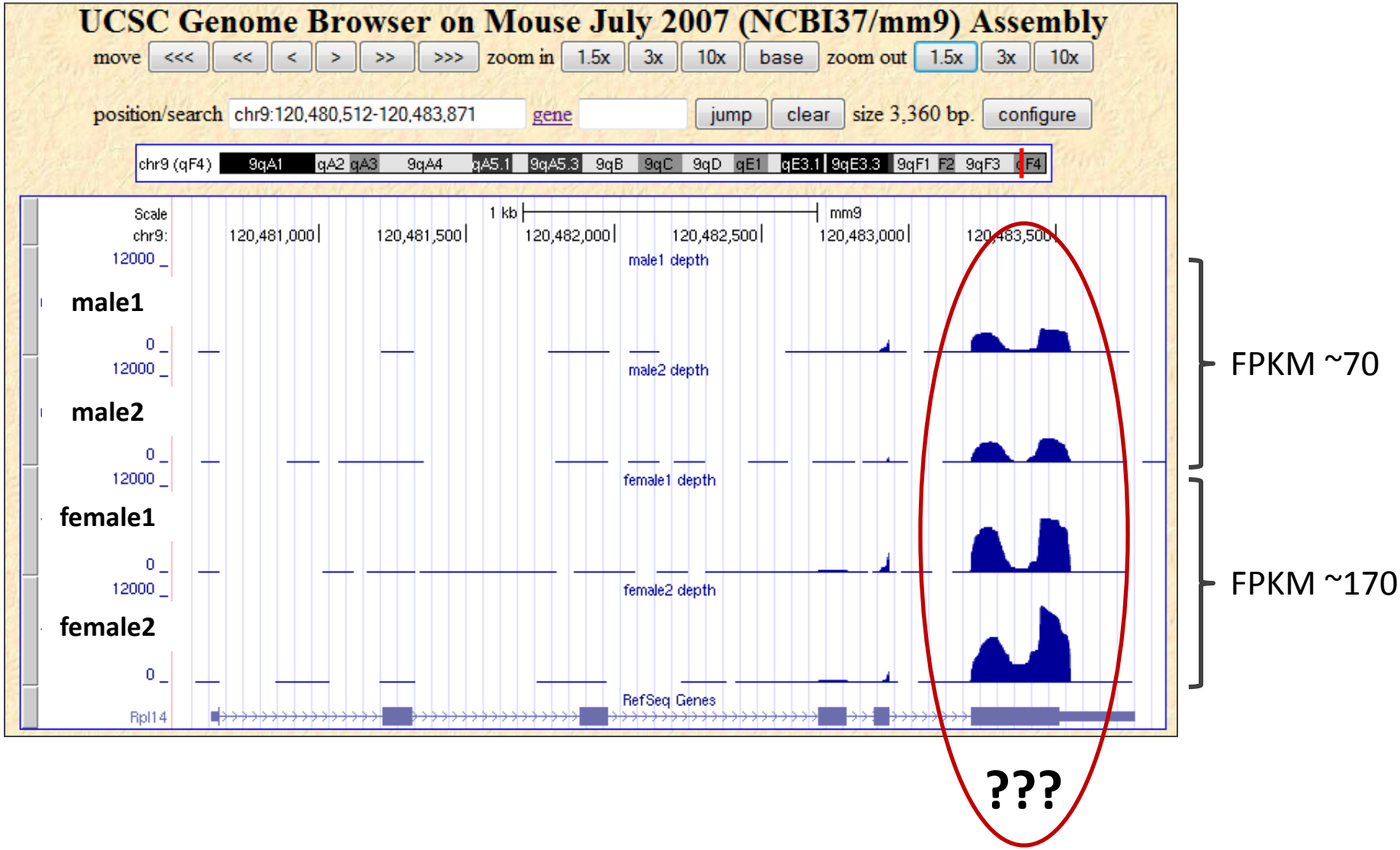
Sort by q-value (i.e. FDR-adjusted P-value) to find most significant differences in gene abundance between males & females. Here are a few somewhat-randomly selected rows…

Gene	Locus	Sample1	Sample2	Status	FPKM 1	FPKM 2	log2 (fold change)	p-value	q-value	Significant
Xist|213742	chrX:100626855-100680296	q1	q2	OK	0.0059	27.1334	12.1752	0	0	yes
Lcn2|16819	chr2:32240156-32243259	q1	q2	OK	10.1613	61.264	2.59196	0	0	yes
Ddx3y|26900	chrY:597157-623056	q1	q2	OK	21.5434	0	-1.8e+308	4.74E-20	5.12E-18	yes
Rpl14|67115	chr9:120480633-120483770	q1	q2	OK	69.7858	166.507	1.2546	1.14E-06	3.07E-05	yes
Insc|233752	chr7:121889279-121993894	q1	q2	OK	5.8597	2.7581	-1.0872	7.23E-05	0.00126	yes

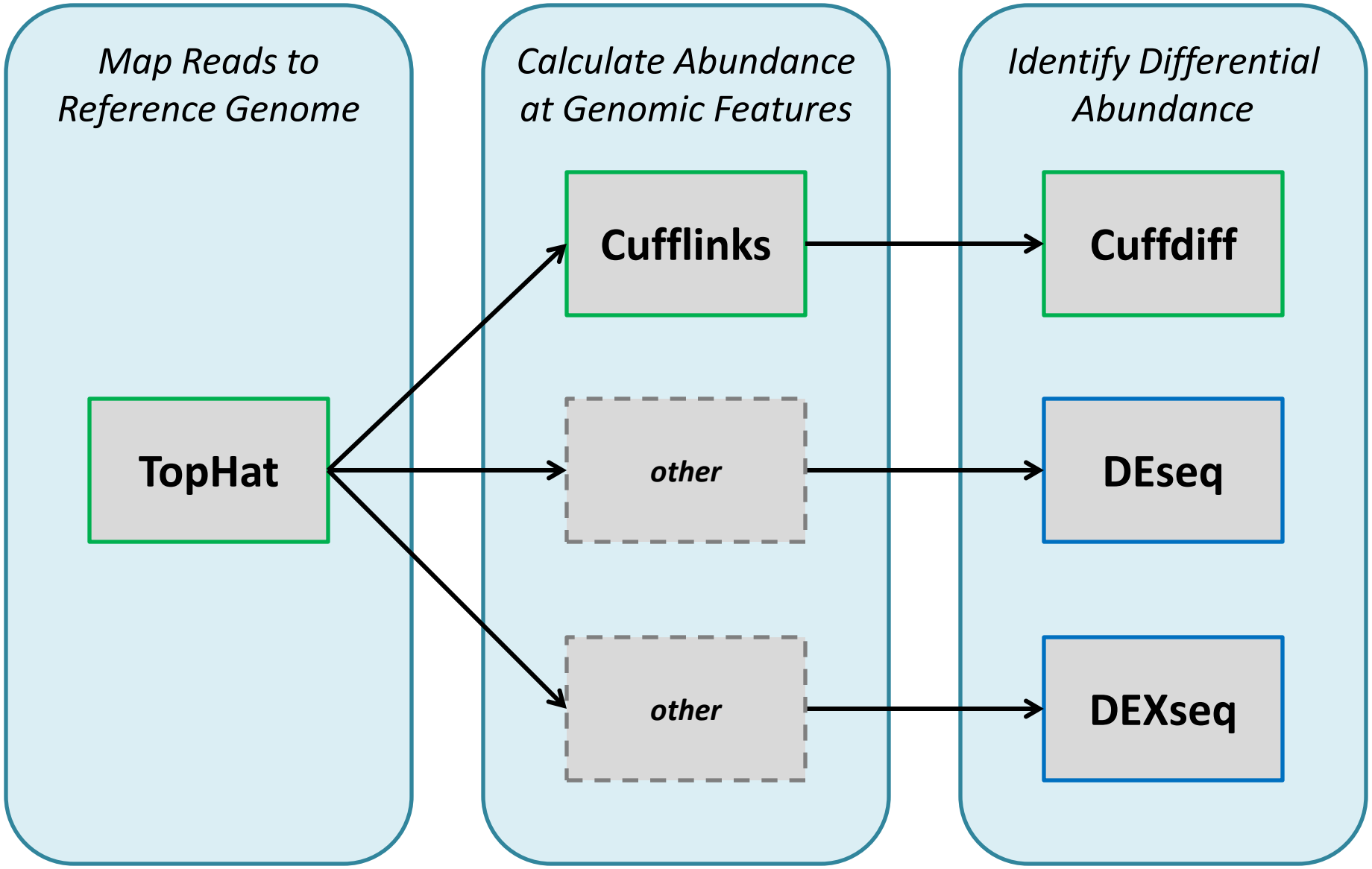
Example: Gene=Lcn2; q-value=0



Example: Gene=Rpl14; q-value = 3.07×10^{-5}



Quantification Workflow Overview



Example RNAseq Analysis: Run DEseq.

Notable differences from Cufflinks/Cuffdiff:

1. All counts are collapsed at the gene level. DEseq doesn't handle anything more complicated, such as estimating different isoforms.

2. All features to be evaluated must be explicitly defined in the input. DEseq can't find novel genes or isoforms.

A) First, re-format your data. You have a collection of SAM/BAM files. DEseq expects you to input a table of read counts per gene per sample, like this:

gene	male1	male2	female1	female2
Aldoart1|353204	12	10	11	17
Aldoart2|79459	0	0	0	10
Aldoa|11674	27244	30883	37535	36595
Aldob|230163	1031063	1308909	1561614	1509842
Aldoc|11676	4124	5266	3919	4381

B) Open an R session. Load your data. Run DEseq. Extract differentially expressed genes according to some defined FDR…

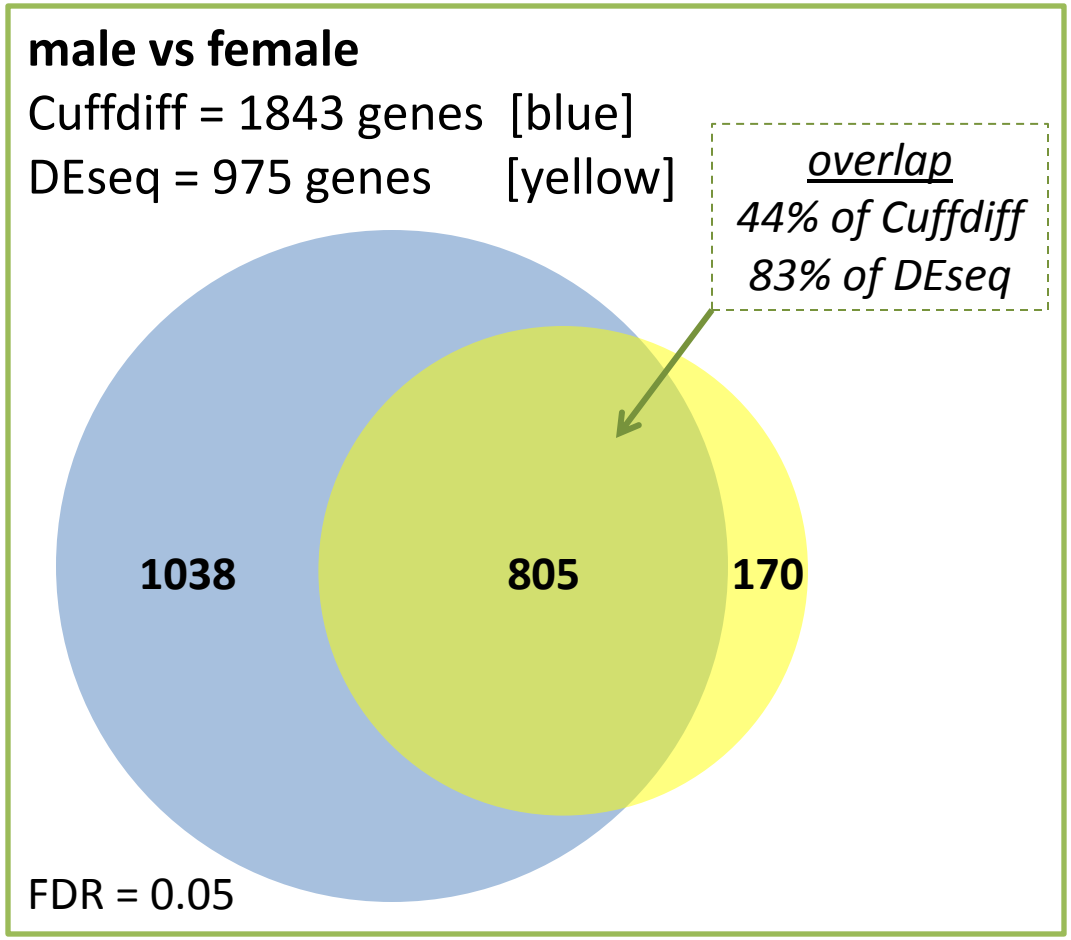
Identifying Differential Abundance from DEseq Output

Example query: What genes are differentially expressed in males & females?

After writing the DEseq output to a text file, it can be loaded in Excel.
Sort by adjusted p-value ("padj") to find most significant differences in gene abundance between males & females. Here are the top few rows…

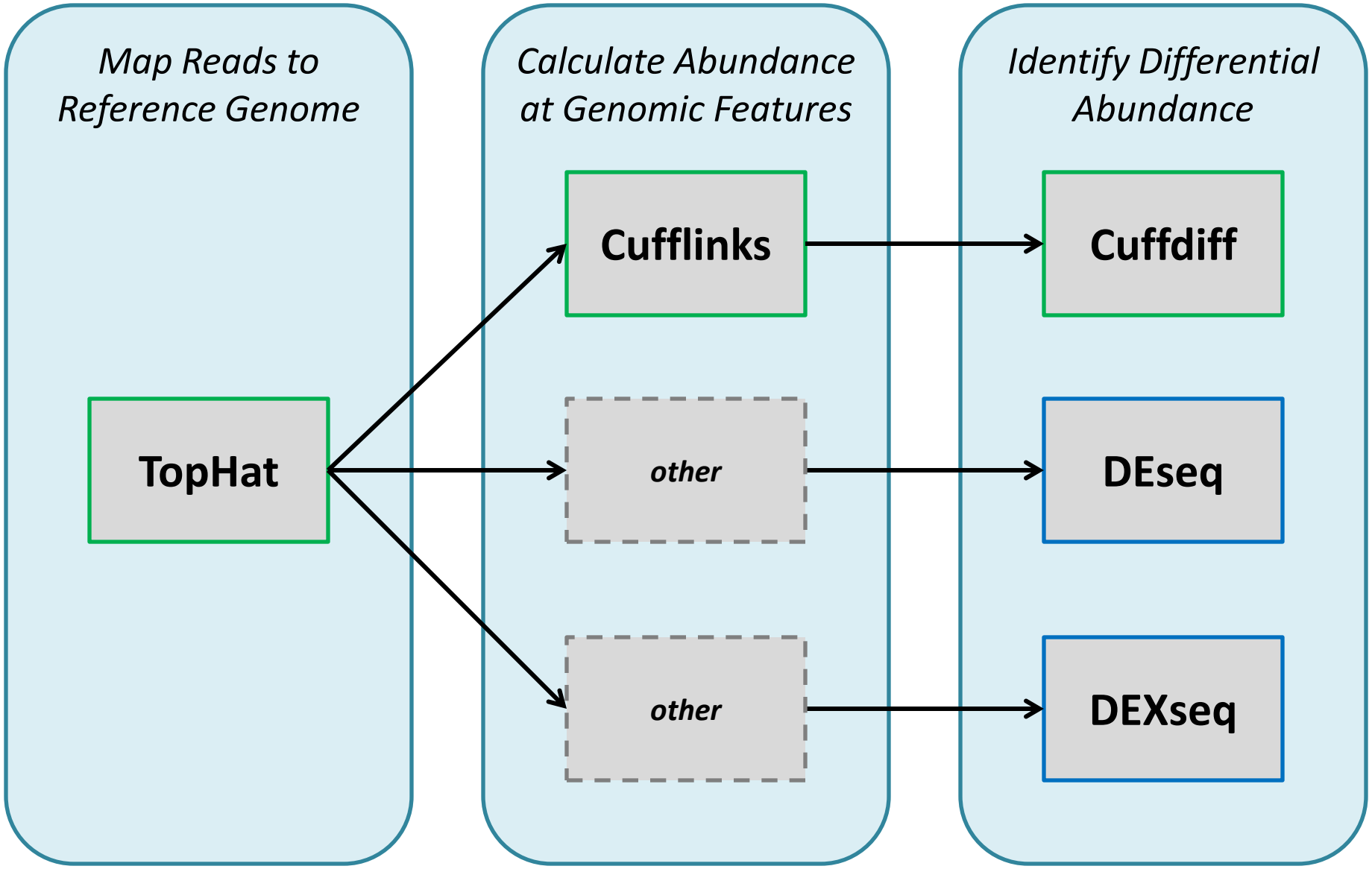
id	baseMean	baseMeanA	baseMeanB	foldChange	log2FoldChange	pval	padj
Sult3a1|57430	69830.831	15.293	139646.368	9131.199	13.157	4.44E-206	8.38E-202
Xist|213742	49518.476	16.928	99020.023	5849.322	12.514	3.13E-193	2.95E-189
Ddx3y|26900	10794.665	21589.331	0	0	-Inf	5.58E-179	3.51E-175
Gm4794|215895	4895.007	0	9790.0135	Inf	Inf	2.49E-144	1.18E-140
Hsd3b5|15496	45776.696	91180.364	373.0285	0.004	-7.933	2.05E-121	7.73E-118
Uty|22290	1721.710	3443.419	0	0	-Inf	2.50E-112	7.86E-109
Kdm5d|20592	1597.569	3195.137	0	0	-Inf	1.38E-109	3.72E-106
Cyp4a12a|277753	77345.359	153271.492	1419.226	0.009	-6.755	4.94E-99	1.17E-95
LOC622167|622167	1174.710	2349.421	0	0	-Inf	3.54E-97	7.42E-94
Acot3|171281	14879.391	272.421	29486.362	108.238	6.758	2.44E-93	4.60E-90

Comparison of Differential Abundance Identified via Cuffdiff and DEseq



Despite that the Cuffdiff & DEseq approaches to identifying differential genes have different underlying statistical models, the overall agreement between the methods is reasonably good.

Quantification Workflow Overview



Example RNAseq Analysis: Run DEXseq.

How is DEseq different than DEXseq?

**** DEseq evaluates independent genomic features – typically used for gene-level analysis.***

**** DEXseq is designed specifically to evaluate differential exon usage.***

A) First, re-format your data. You have a collection of SAM/BAM files. DEXseq expects you to give it read counts per exon per sample. This table demonstrates the necessary information (although in the wrong format)...

gene	exon	chr	start	end	strand	male1	male2	female1	female2
Abhd8|64296	E001	chr8	73,980,599	73,981,276	-	130	107	177	157
Abhd8|64296	E002	chr8	73,981,937	73,982,153	-	42	47	49	58
Abhd8|64296	E003	chr8	73,982,277	73,982,447	-	41	36	26	30
Abhd8|64296	E004	chr8	73,985,145	73,985,871	-	86	62	32	41
Abhd8|64296	E005	chr8	73,987,365	73,987,556	-	43	16	3	13

B) Open an R session. Load your data. Run DEXseq to compare exon usage across samples.

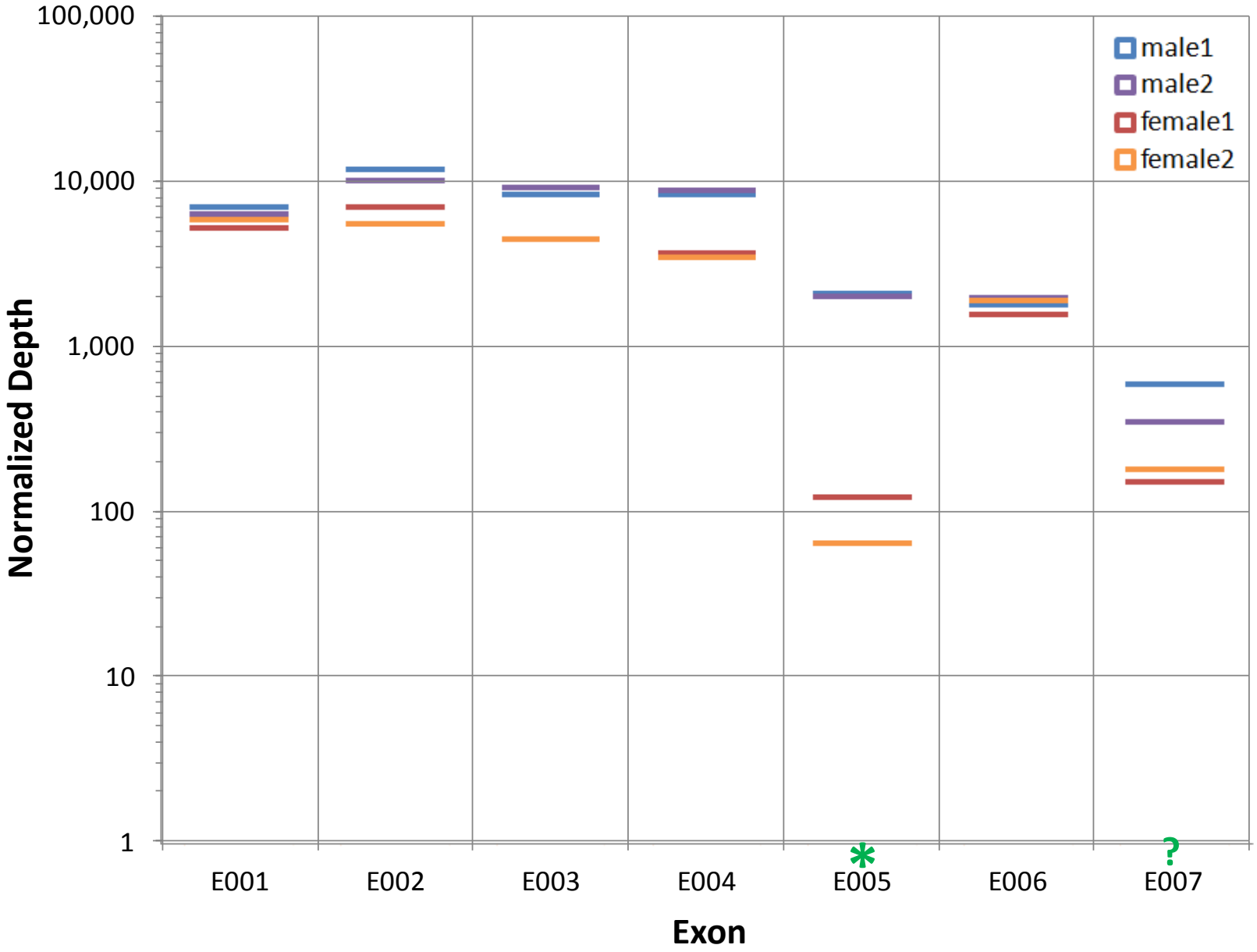
Identifying Differential Abundance from DEXseq Output

Example query: What exons are differentially expressed in males & females?

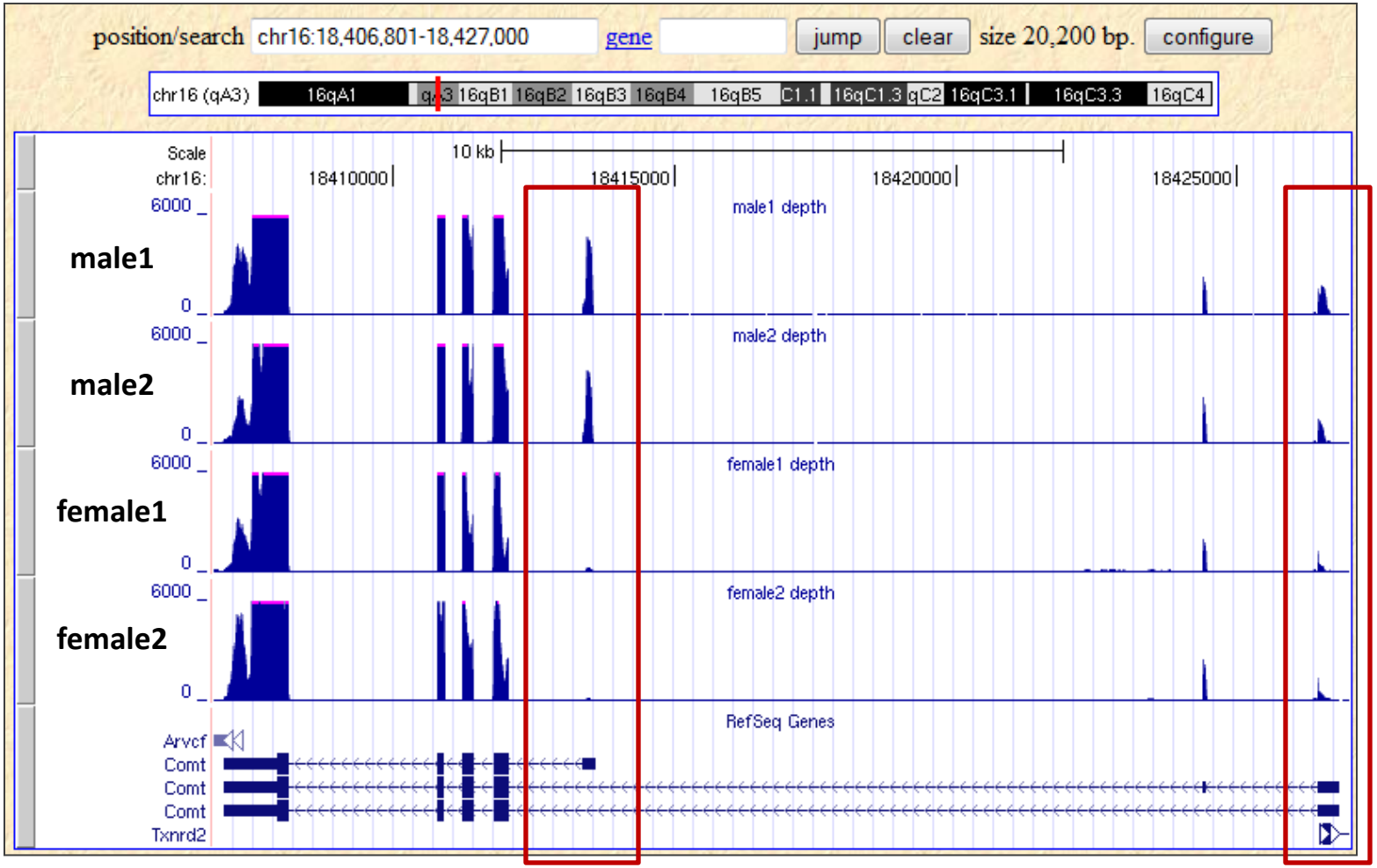
For each exon in the reference set, DEXseq reports a P-value for significance of differential abundance. For example…

	geneID	exonID	testable	adj P-value	chr	start	end	strand	transcripts
	Comt|12846	E001	TRUE	0.087	chr16	18,406,975	18,408,133	-	NM_007744; NM_001111062; NM_001111063
	Comt|12846	E002	TRUE	0.850	chr16	18,410,776	18,410,907	-	NM_007744; NM_001111062; NM_001111063
	Comt|12846	E003	TRUE	0.641	chr16	18,411,220	18,411,413	-	NM_007744; NM_001111062; NM_001111063
	Comt|12846	E004	TRUE	0.143	chr16	18,411,783	18,412,050	-	NM_007744; NM_001111062; NM_001111063
*	Comt|12846	E005	TRUE	1.07E-09	chr16	18,413,359	18,413,587	-	NM_001111063
	Comt|12846	E006	TRUE	0.194	chr16	18,424,397	18,424,453	-	NM_001111062
?	Comt|12846	E007	TRUE	0.014	chr16	18,426,449	18,426,809	-	NM_007744; NM_001111062

Example: Gene=Comt



Example: Gene=Comt



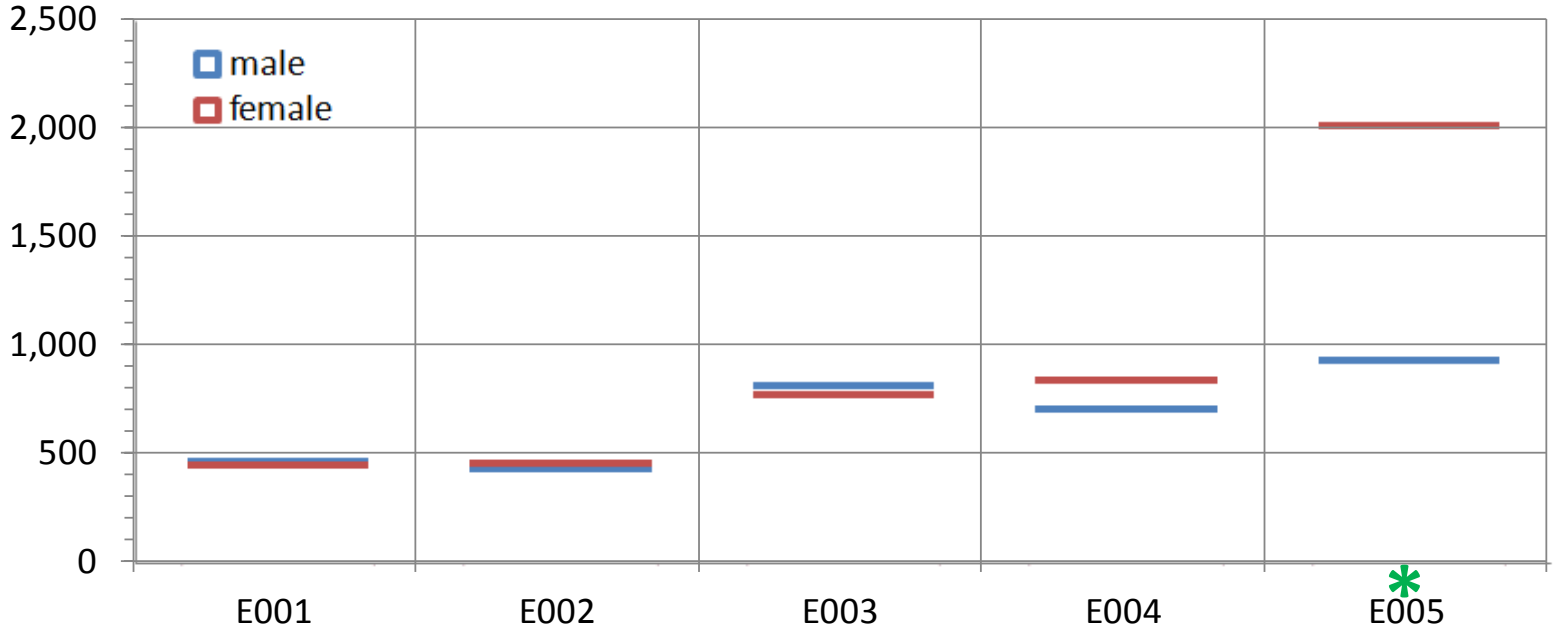
Fold change: ~20x
adj P-value: ~10^{-9}
[E005]

Fold change: ~2.5x
adj P-value: 0.014
[E007]

Identifying Differential Abundance from DEXseq Output

Not all differentially expressed exons are as easy to interpret…

geneID	exonID	testable	adj P-value	chr	start	end	strand	transcripts
Rgs16|19734	E001	TRUE	0.309	chr1	155,587,483	155,587,642	+	NM_011267
Rgs16|19734	E002	TRUE	0.698	chr1	155,587,993	155,588,103	+	NM_011267
Rgs16|19734	E003	TRUE	0.388	chr1	155,588,750	155,588,811	+	NM_011267
Rgs16|19734	E004	TRUE	0.894	chr1	155,589,093	155,589,259	+	NM_011267
Rgs16|19734	E005	TRUE	2.91E-05	chr1	155,590,764	155,592,598	+	NM_011267



Identification of Novel Genes & Isoforms By Cufflinks

If Cufflinks is run without a set of reference annotations (genes & transcripts), it will attempt to identify transcripts from the mapped sequence data.

Output files:

genes.fpkm_tracking

isoforms.fpkm_tracking

transcripts.gtf

These files include the FPKM calculated for each gene or each transcript/isoform that is identified by cufflinks.

Genomic locations of all identified transcripts in a standard format (GTF).

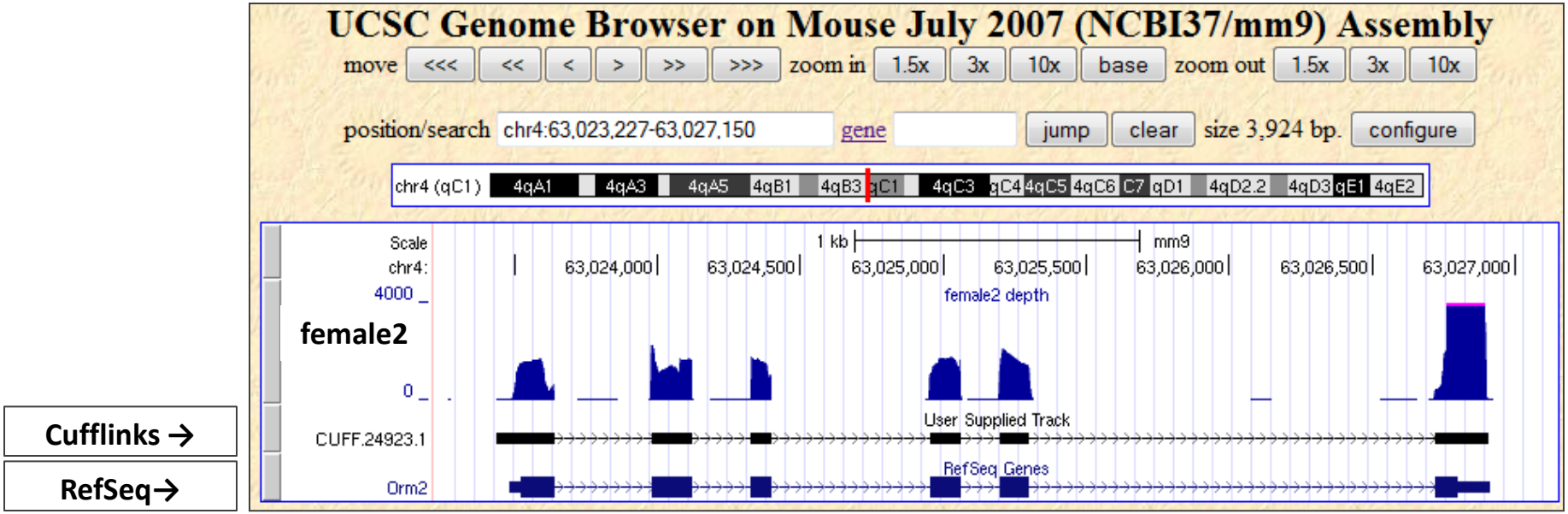
Each entry has some mapped-read support according to the cufflinks algorithm.

Identify which features are novel by comparing this file against a set of known reference features…

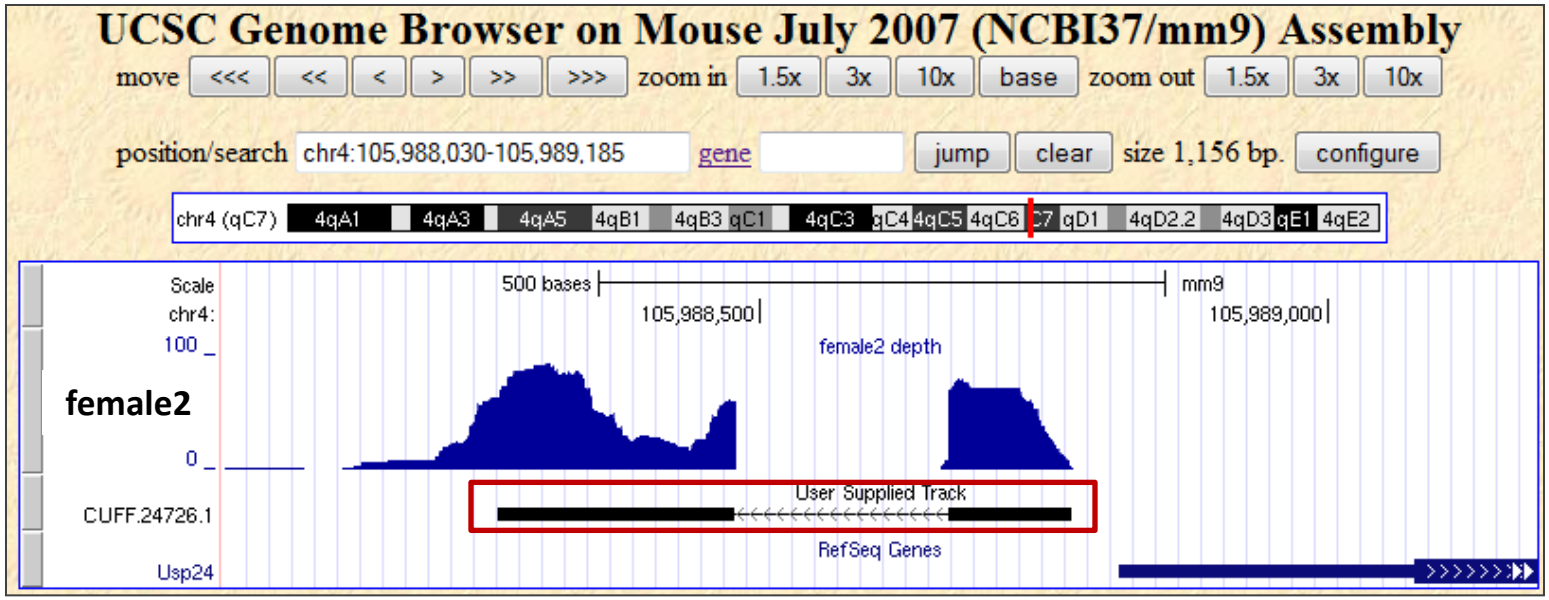
Cufflinks (Sometimes) Finds Similar Gene Model to RefSeq

For many loci, Cufflinks is able to reconstruct a gene model very similar to RefSeq.

Example: Gene=Orm2

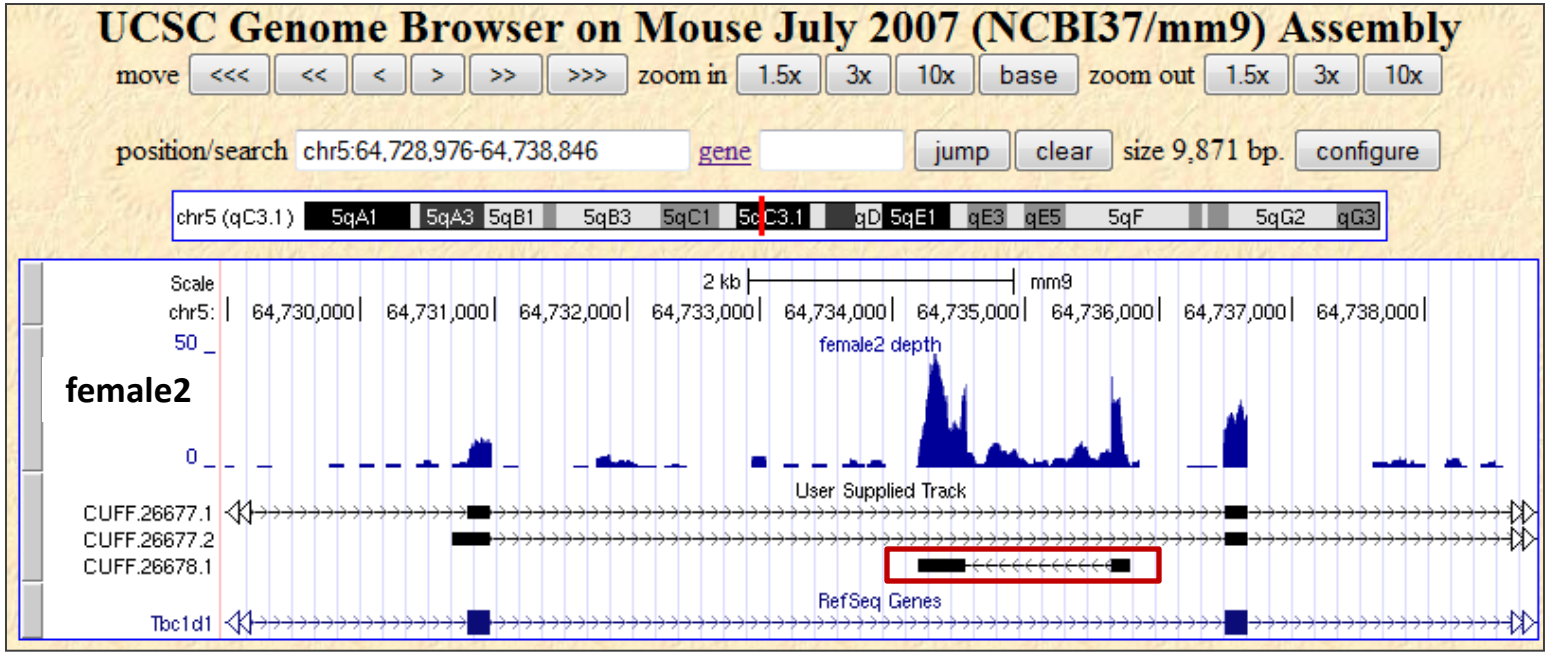


Cufflinks (Sometimes) Finds Potential Novel Genes



Cufflinks →

RefSeq→



Cufflinks →

RefSeq→

Tools Discussed Today

FastQC

One of many available options for performing QC evaluations. *(Bonus: You can run this in Galaxy. Check out Tom Randall's Galaxy course for a demo!)*

TopHat

The current gold standard for mapping RNAseq reads to a reference genome.

Cufflinks/Cuffdiff

A popular quantification tool, part of the same tool suite as TopHat.

Does not require reference annotations (i.e. can find potentially novel genes & isoforms).

DEseq/DEXseq

Another popular quantification tool. These are 2 related R packages.

Considered by many to be statistically superior to Cufflinks/Cuffdiff.

Requires user-defined genomic features, so won't find anything novel.