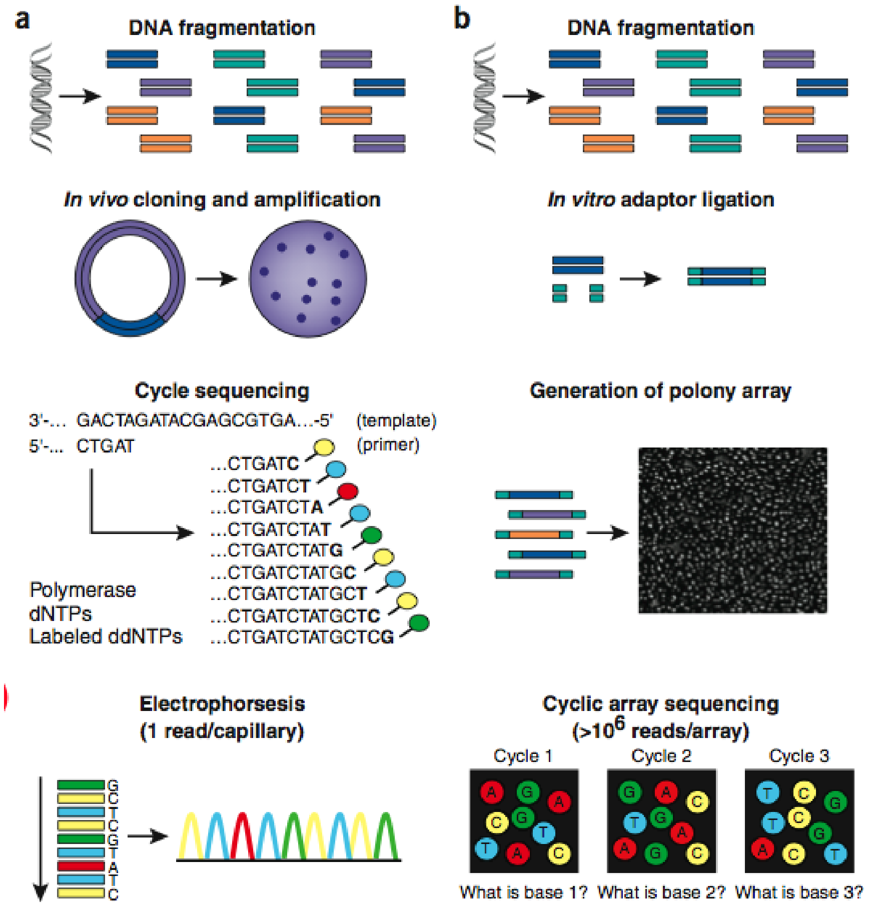
RNA-seq

Manpreet S. Katari

Evolution of Sequence Technology



Next-generation DNA sequencing

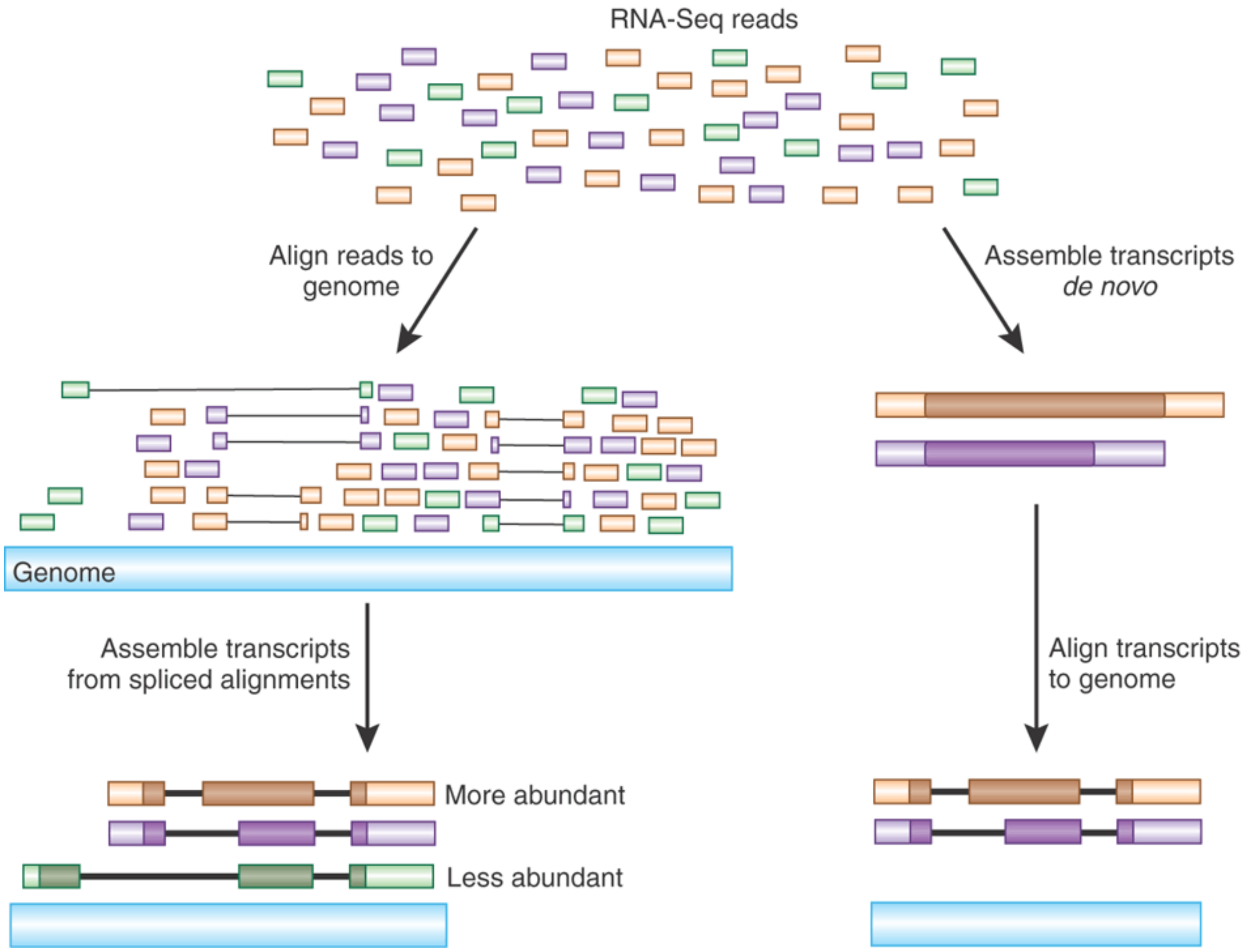
Jay Shendure[1] & Hanlee Ji[2]

NATURE BIOTECHNOLOGY VOLUME 26 NUMBER 10 OCTOBER 2008

Advancing RNA-Seq analysis

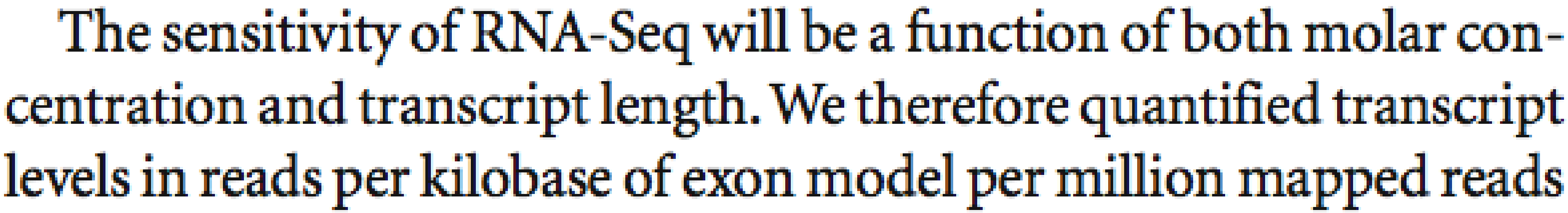
Brian J Haas & Michael C Zody

Nature Biotechnology **28**, 421–423 (2010) | doi:10.1038/nbt0510-421


RNA-Seq reads
Align reads to genome
Assemble transcripts *de novo*
Genome
Assemble transcripts from spliced alignments
Align transcripts to genome
More abundant
Less abundant

Normalizing the Data

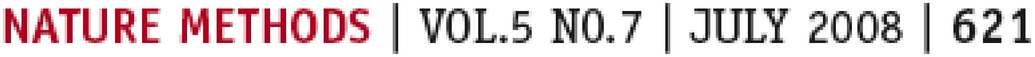
- RPKM (Reads per Kilobase of exons per million reads)



The sensitivity of RNA-Seq will be a function of both molar concentration and transcript length. We therefore quantified transcript levels in reads per kilobase of exon model per million mapped reads

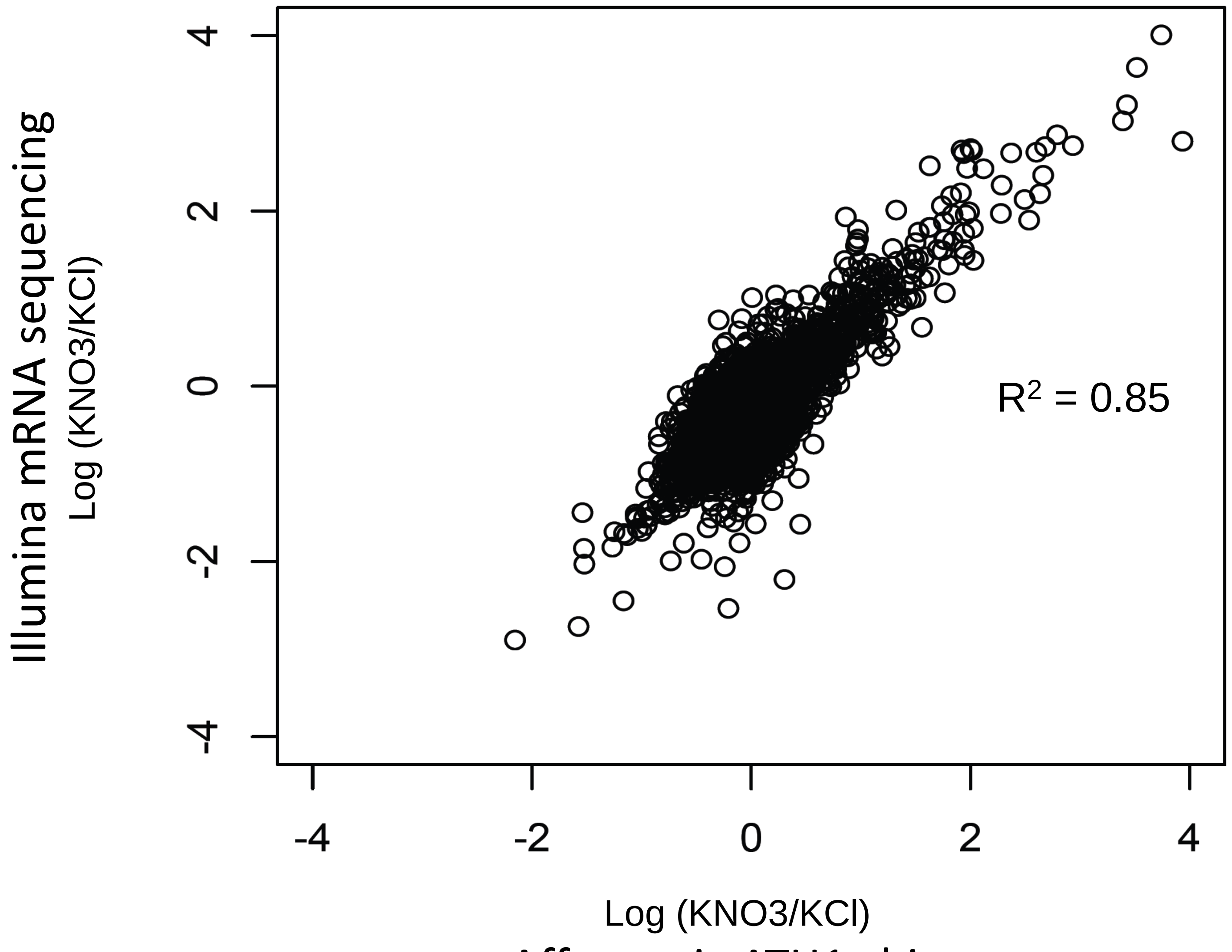
$$\text{Score} = \frac{R}{NT}$$

R = # of unique reads for the gene
N = Size of the gene (sum of exons / 1000)
T = total number of reads in the library mapped to the genome / 1,000,000

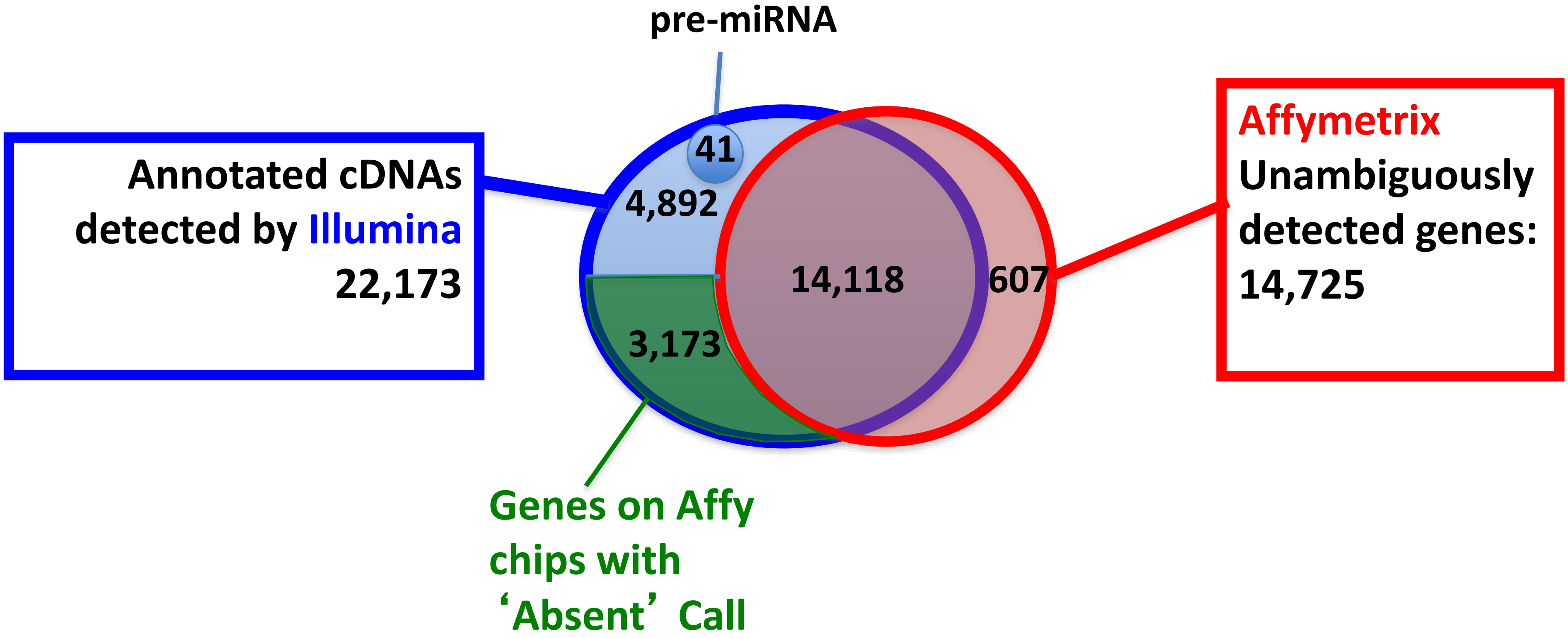


NATURE METHODS | VOL.5 NO.7 | JULY 2008 | 621

N-regulation of mRNA: Illumina vs ATH1



Detection of Arabidopsis Genes



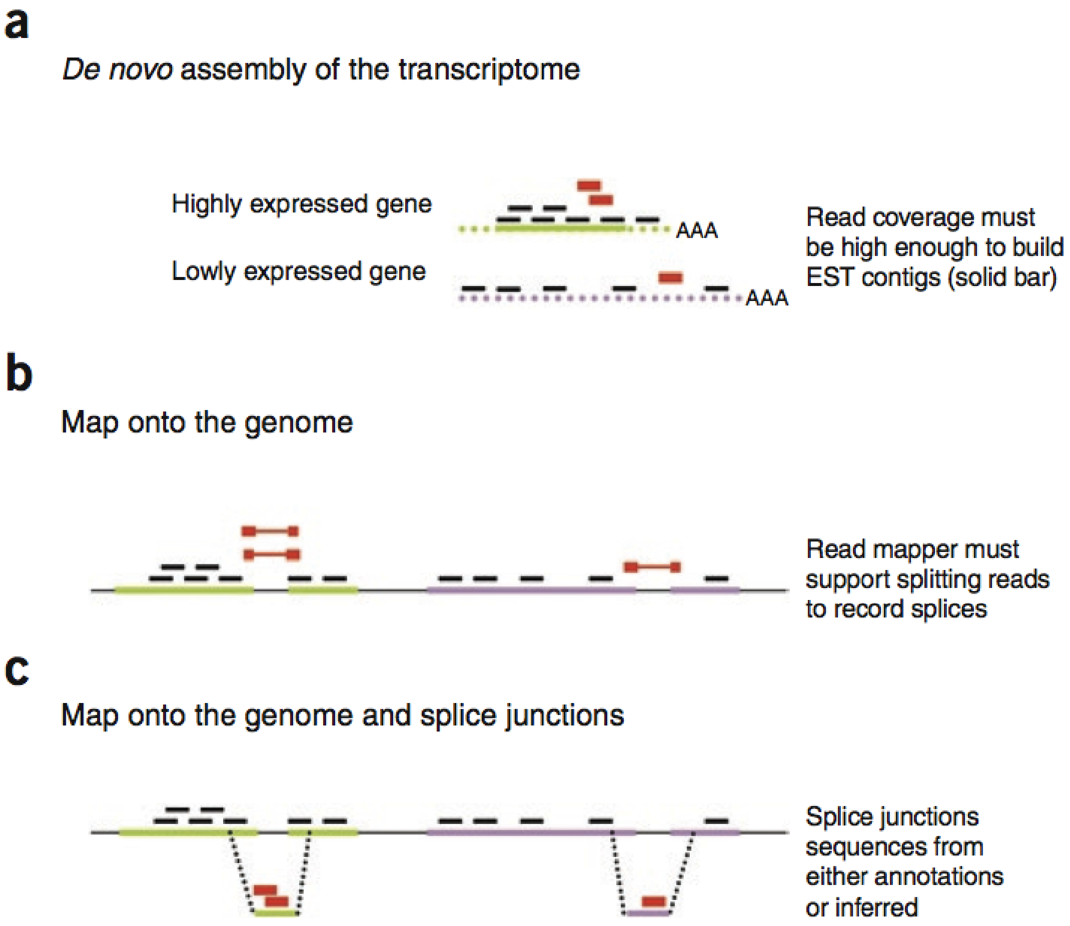
pre-miRNA

Annotated cDNAs detected by Illumina 22,173

Affymetrix Unambiguously detected genes: 14,725

Genes on Affy chips with ‘Absent’ Call

RNA-seq provides even more



a

De novo assembly of the transcriptome

Read coverage must be high enough to build EST contigs (solid bar)

b

Map onto the genome

Read mapper must support splitting reads to record splices

c

Map onto the genome and splice junctions

Splice junctions sequences from either annotations or inferred

RNA-seq pipeline

Manpreet S. Katari

The basic workflow

1. Evaluate the quality of the sequences
    a. Use fastqc to asses quality of sequence
2. Trim low quality sequences
    a. Use fastx tool kit
3. Map the reads to the Genome
    a. Build the bowtie2 database
    b. Run the alignment using tophat2
4. Link mapped reads into genes and calculate normalized expression values
    a. Use cufflinks to determine normalized values of each run.
5. Compare samples to determine differentially expressed genes.
    a. Use cuffdiff to compare the different samples and identify differentially expressed genes.

Processing RNA-seq reads (Filter)

- Remove not so interesting RNA molecules
  - Majority of the RNA molecules in the cell are ribosomal rna.
- Low complexity sequences
  - For example PolyA sequences.
- Adapter sequences
  - Occasionally some of the reads can contain adapter sequences.
- Illumina reads have tendency to have poor quality reads in the 3'
  - Trim reads on either end and also based on quality.

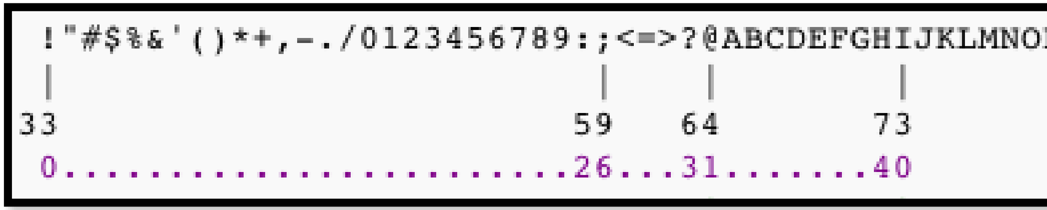
Fastq format

Read Identifier

Read Sequence

Read Sequence Quality

```
@HYYD8:00025:00048
GGGTTTTCAGGGGAAAAGAAAA
+
DD7BBB7BBBBA5@?=/6666)
@HYYD8:00026:00046
TCCCTTTGGT
+
BBC=BB3737
@HYYD8:00027:00046
AAAAAAAAA
+
:DBBCCBB&
@HYYD8:00027:00049
CTGCAACGTTGACCCAT
+
@??BB<???-3344*34
@HYYD8:00029:00045
ACGATTGGTTTTTTAGTTGGTTGGGTTTGGTTTTTTTTTGGGTTGG
+
8774=;?;AA?AA*A@A;?:>-67+55+5:@BBCCCCC&CB<C<?-
```



```
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNO
 |                         |    |        |
33                        59   64       73
 0........................26...31.......40
```

Module environment review

- To look at the different modules available:

  -
    ```
    module avail
    ```
- To load a module

  -
    ```
    module load fastqc
    ```
- To get a list of modules already loaded

  -
    ```
    module list
    ```
- To remove or unload a module

  -
    ```
    module unload fastqc
    ```
- To get help on fastqc

  -
    ```
    fastqc -h
    ```

1. Perform Quality control

- We will use the Fastqc package to evaluate the quality of our sequences.
- http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

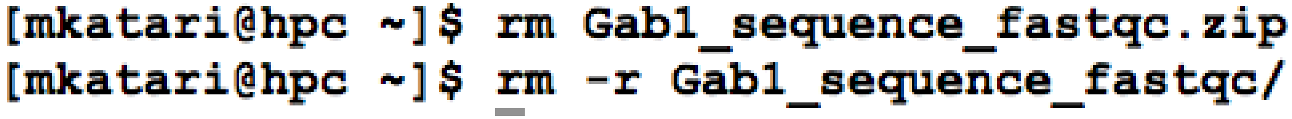
```
module load fastqc/0.10.1

fastqc sequence.fastq
```

- Transfer the folder to your local machine to view results.
  - On the pc we used the sftp on mobaXterm
    - Another option is WinSCP (http://winscp.net/eng/download.php)
  - On the mac you can use cyberduck or scp commad

- Extract the folder and open the fastqc_report.html

Cleaning up

- Once you have the files on your computer and you don’t need it on the server simply delete them.



```
[mkatari@hpc ~]$ rm Gab1_sequence_fastqc.zip
[mkatari@hpc ~]$ rm -r Gab1_sequence_fastqc/
```

The –r option allows you to delete recursively all files and directories below the one provided.

2. Trimming the reads

- Sequences generated from illumina platform tend to have lower quality sequences specially at the 3' end.
- Since our sequence alignment algorithms are looking for nearly exact match, we want to trim the sequence from 5’ and 3’ end.
- Our sequence is 50bp, so let's trim 5 from 5’ and stop at base 40.
- We also noticed that
- We will use a software available for free called fastx
  - ○ http://hannonlab.cshl.edu/fastx_toolkit/

```
module load fastx_toolkit/0.0.13
fastx_trimmer -f 5 sequence.fastq -l 40 -o
   sequence.trimmed.fastq
fastx_clipper -a ATCGTATGCCGTCTTCTGCTTG -l 25 -I
   sequence.trimmed.fastq -o  sequence_trimmed_clipped.fastq
```

Aligning Short reads

New Algorithms for short sequences

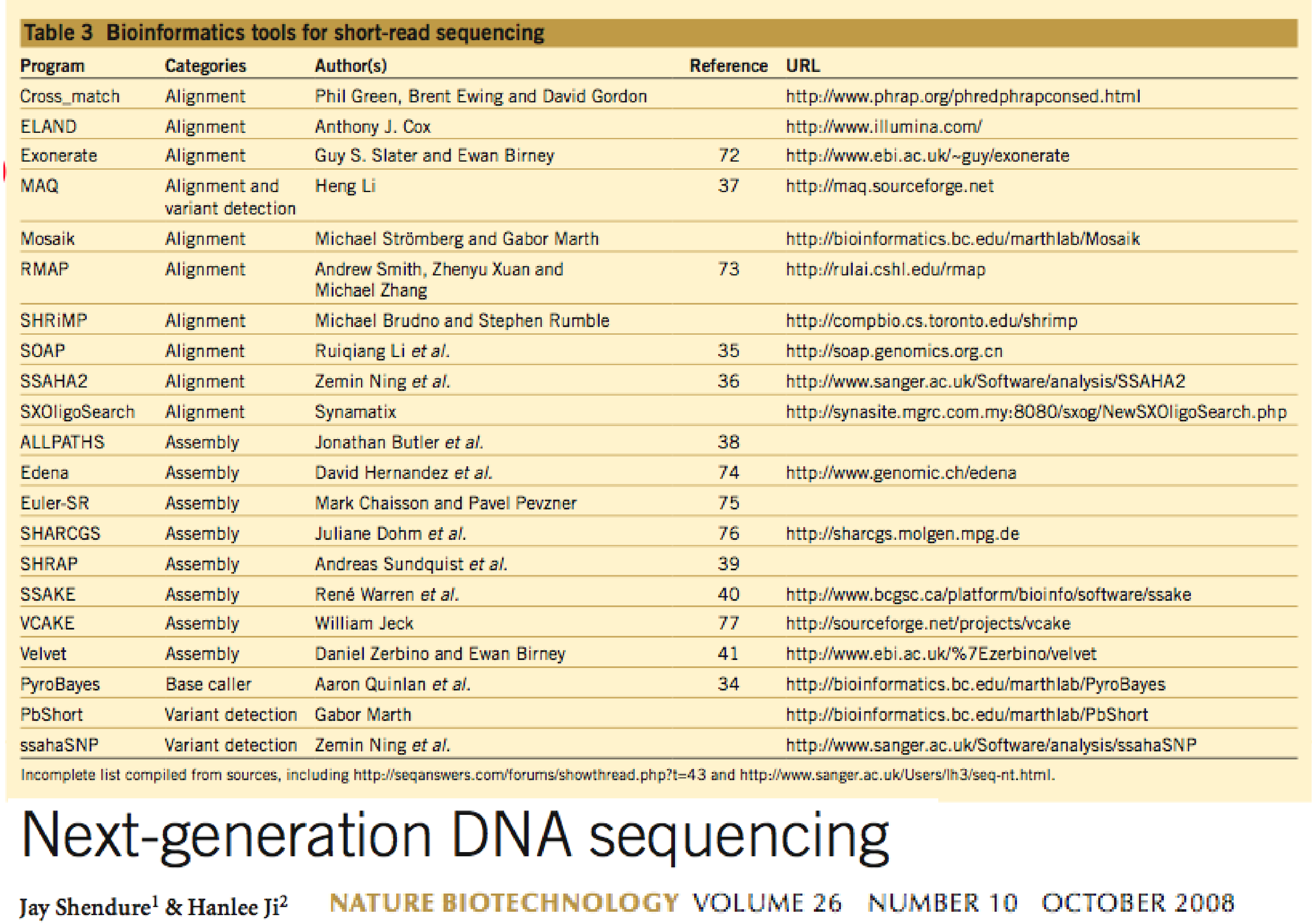


Table 3 Bioinformatics tools for short-read sequencing

Program	Categories	Author(s)	Reference	URL
Cross_match	Alignment	Phil Green, Brent Ewing and David Gordon		http://www.phrap.org/phredphrapconsed.html
ELAND	Alignment	Anthony J. Cox		http://www.illumina.com/
Exonerate	Alignment	Guy S. Slater and Ewan Birney	72	http://www.ebi.ac.uk/~guy/exonerate
MAQ	Alignment and variant detection	Heng Li	37	http://maq.sourceforge.net
Mosaik	Alignment	Michael Strömberg and Gabor Marth		http://bioinformatics.bc.edu/marthlab/Mosaik
RMAP	Alignment	Andrew Smith, Zhenyu Xuan and Michael Zhang	73	http://rulai.cshl.edu/rmap
SHRiMP	Alignment	Michael Brudno and Stephen Rumble		http://compbio.cs.toronto.edu/shrimp
SOAP	Alignment	Ruiqiang Li *et al.*	35	http://soap.genomics.org.cn
SSAHA2	Alignment	Zemin Ning *et al.*	36	http://www.sanger.ac.uk/Software/analysis/SSAHA2
SXOligoSearch	Alignment	Synamatix		http://synasite.mgrc.com.my:8080/sxog/NewSXOligoSearch.php
ALLPATHS	Assembly	Jonathan Butler *et al.*	38	
Edena	Assembly	David Hernandez *et al.*	74	http://www.genomic.ch/edena
Euler-SR	Assembly	Mark Chaisson and Pavel Pevzner	75	
SHARCGS	Assembly	Juliane Dohm *et al.*	76	http://sharcgs.molgen.mpg.de
SHRAP	Assembly	Andreas Sundquist *et al.*	39	
SSAKE	Assembly	René Warren *et al.*	40	http://www.bcgsc.ca/platform/bioinfo/software/ssake
VCAKE	Assembly	William Jeck	77	http://sourceforge.net/projects/vcake
Velvet	Assembly	Daniel Zerbino and Ewan Birney	41	http://www.ebi.ac.uk/%7Ezerbino/velvet
PyroBayes	Base caller	Aaron Quinlan *et al.*	34	http://bioinformatics.bc.edu/marthlab/PyroBayes
PbShort	Variant detection	Gabor Marth		http://bioinformatics.bc.edu/marthlab/PbShort
ssahaSNP	Variant detection	Zemin Ning *et al.*		http://www.sanger.ac.uk/Software/analysis/ssahaSNP

Incomplete list compiled from sources, including http://seqanswers.com/forums/showthread.php?t=43 and http://www.sanger.ac.uk/Users/lh3/seq-nt.html.

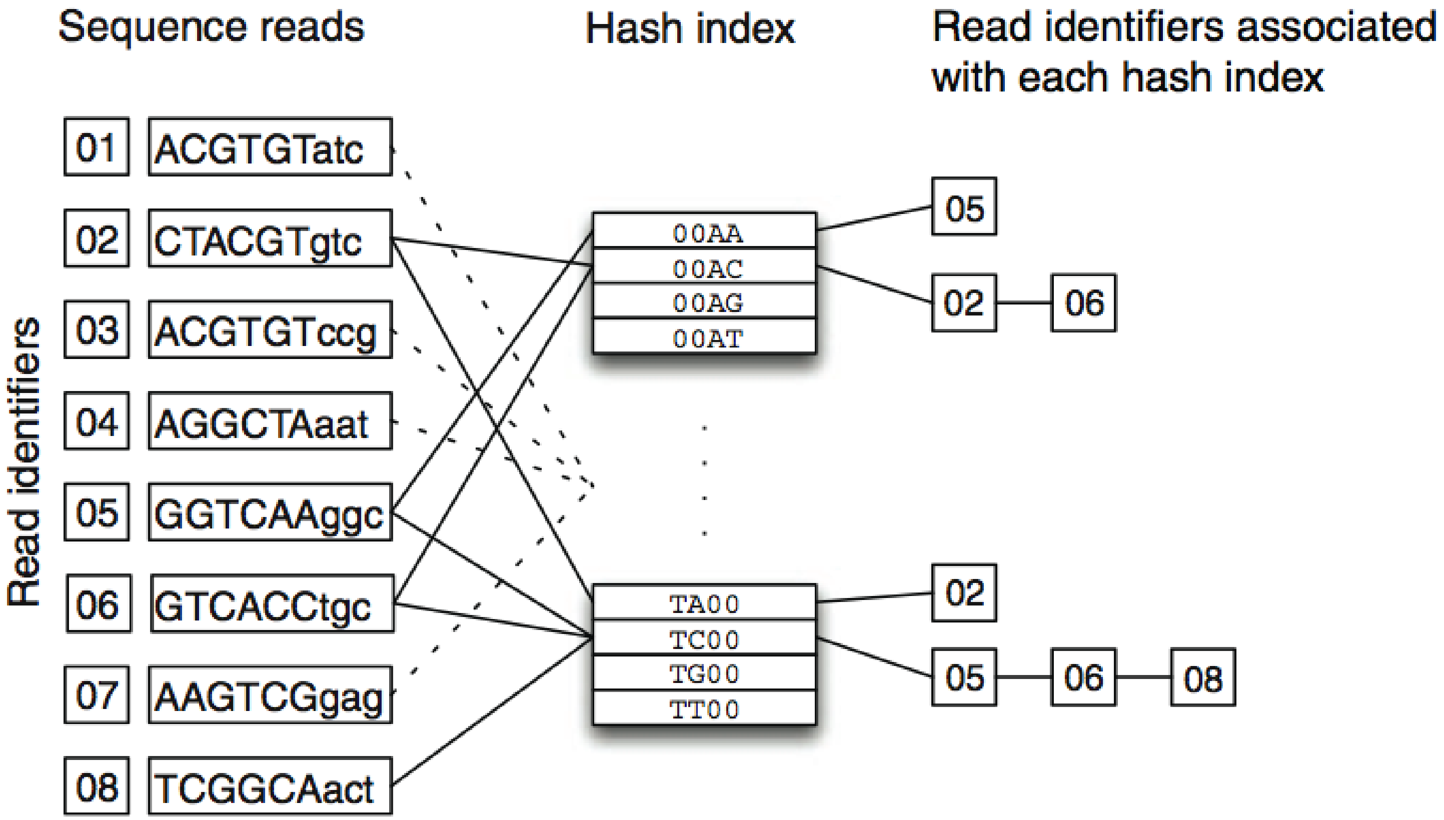
Next-generation DNA sequencing

Jay Shendure[1] & Hanlee Ji[2] NATURE BIOTECHNOLOGY VOLUME 26 NUMBER 10 OCTOBER 2008

Two main types of alignment methods

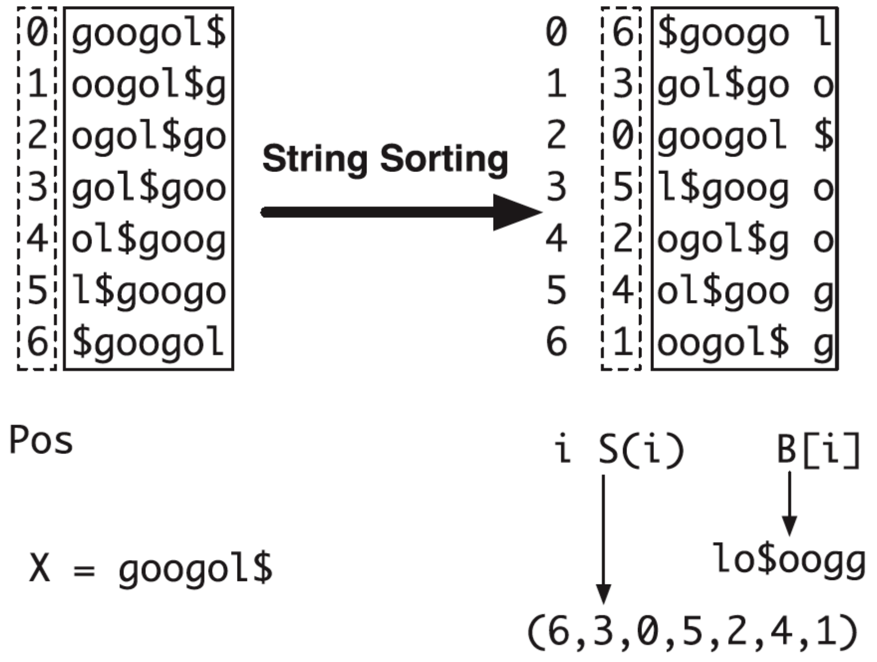
- Hash-table based
- Burrows and Wheeler Transformation
- Both can be applied to Illumina and Solid
- Both start with different heuristics to reduce the search space but then finally use a more accurate alignment method like Smith Waterman.

Hash Table (BLAT)



NATURE METHODS SUPPLEMENT | VOL.6 NO.11s | NOVEMBER 2009

Burrows Wheeler Transformation



Pos	
0	googol$
1	oogol$g
2	ogol$go
3	gol$goo
4	ol$goog
5	l$googo
6	$googol

i	S(i)		B[i]
0	6	$googo	l
1	3	gol$go	o
2	0	googol	$
3	5	l$goog	o
4	2	ogol$g	o
5	4	ol$goo	g
6	1	oogol$	g

Fig. 2. Constructing suffix array and BWT string for X• googol$. String X is circulated to generate seven strings, which are then lexicographically sorted. After sorting, the positions of the first symbols form the suffix array (6•3•0•5•2•4•1) and the concatenation of the last symbols of the circulated strings gives the BWT string lo$oogg.

Li and Durbin (2009) *Bioinformatics*

Which is better ?

- BWA is about 10x faster then hash-based methods and takes less memory.
- BWA is less sensitive. Based on the query size it can only allow a given number of mismatches
  - For example for 100bp max of 5 mismatch.

Mapping Reads from RNA molecules

- What is the advantage of mapping reads from RNA to the genome sequenced instead of a database of all predicted RNA molecules?
  - We are not depending on the quality of annotation.
  - We are not assuming that we know about all of the RNA molecules in the cell.
- How can we find reads mapping to spliced junctions?
  - Create a separate database of all possible spliced junctions
  - Split reads in half and map them separately.

Bowtie
&
TopHat

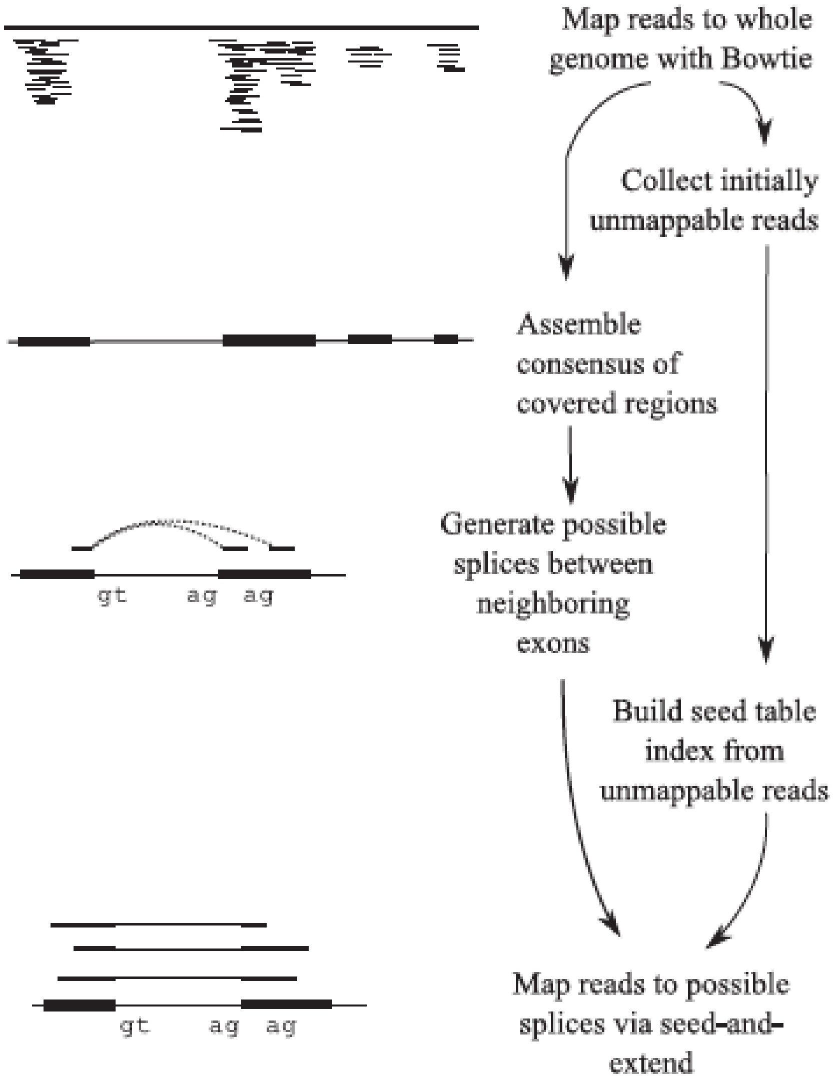

Map reads to whole genome with Bowtie
Collect initially unmappable reads
Assemble consensus of covered regions
Generate possible splices between neighboring exons
gt
ag
ag
Build seed table index from unmappable reads
Map reads to possible splices via seed-and-extend
gt
ag
ag

Fig. 1. The TopHat pipeline. RNA-Seq reads are mapped against the whole reference genome, and those reads that do not map are set aside. An initial consensus of mapped regions is computed by Maq. Sequences flanking potential donor/acceptor splice sites within neighboring regions are joined to form potential splice junctions. The IUM reads are indexed and aligned to these splice junction sequences.

Langmead B, Salzberg S. [Fast gapped-read alignment with Bowtie 2](). *[Nature Methods]()*. 2012, 9:357-359.

3. Mapping the reads

To align our sequences to the genome we will use the Bowtie-Tophat algorithm discussed in class

http://bowtie-bio.sourceforge.net/index.shtml

1. Building the database
   a. In order to use bowtie and tophat for our analysis we have to first create the database.
   b. The following command will create a database in your current directory

```
module load bowtie2/2.2.2
bowtie2-build /home/mkatari/Arabidopsis.fa Arabidopsis
```

3. Mapping the reads

2. Run the alignment

```
module load tophat2/2.0.11
tophat2 -i 20 -I 12000 -o tophat_output \
   /home/mkatari/nitrogen/Arabidopsis \
   sequence_trimmed_clipped.fastq
```

-i = minimum intron size

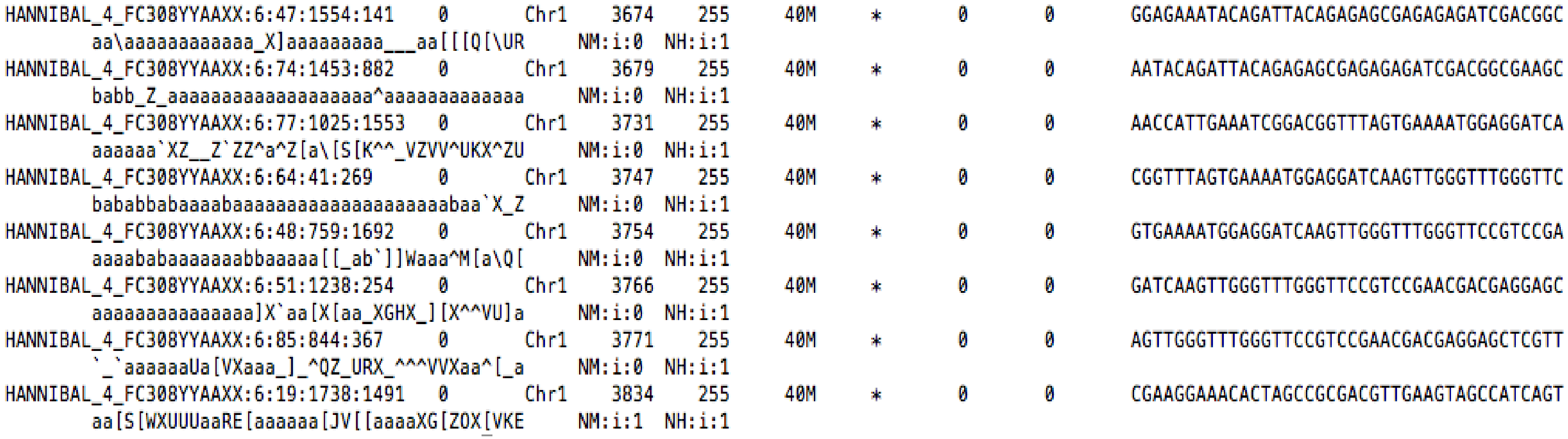
-I = maximum intron size

-o = output directory

Database

Query file

Tophat result: sam file



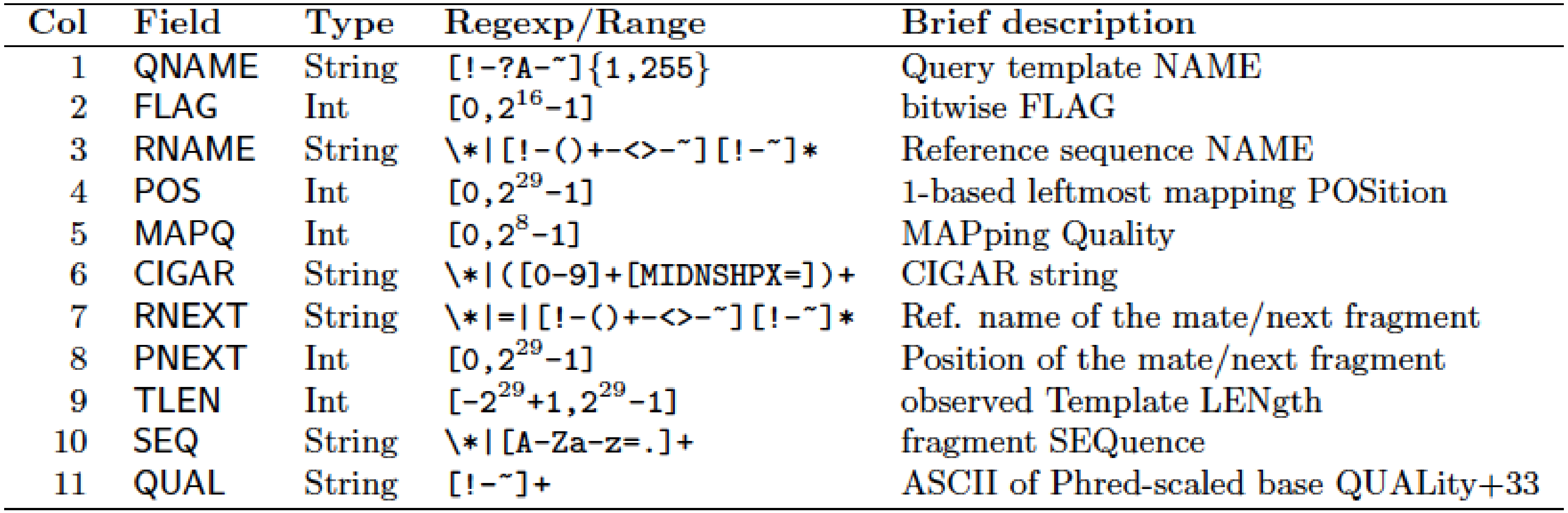
```
HANNIBAL_4_FC308YYAAXX:6:47:1554:141    0       Chr1    3674    255     40M     *       0       0       GGAGAAATACAGATTACAGAGAGCGAGAGAGATCGACGGC
        aa\aaaaaaaaaaaa_X]aaaaaaaaa___aa[[[Q[\UR     NM:i:0  NH:i:1
HANNIBAL_4_FC308YYAAXX:6:74:1453:882    0       Chr1    3679    255     40M     *       0       0       AATACAGATTACAGAGAGCGAGAGAGATCGACGGCGAAGC
        babb_Z_aaaaaaaaaaaaaaaaaaa^aaaaaaaaaaaaa     NM:i:0  NH:i:1
HANNIBAL_4_FC308YYAAXX:6:77:1025:1553   0       Chr1    3731    255     40M     *       0       0       AACCATTGAAATCGGACGGTTTAGTGAAAATGGAGGATCA
        aaaaaa`XZ__Z`ZZ^a^Z[a\[S[K^^_VZVV^UKX^ZU     NM:i:0  NH:i:1
HANNIBAL_4_FC308YYAAXX:6:64:41:269      0       Chr1    3747    255     40M     *       0       0       CGGTTTAGTGAAAATGGAGGATCAAGTTGGGTTTGGGTTC
        bababbabaaaabaaaaaaaaaaaaaaaaaaaabaa`X_Z     NM:i:0  NH:i:1
HANNIBAL_4_FC308YYAAXX:6:48:759:1692    0       Chr1    3754    255     40M     *       0       0       GTGAAAATGGAGGATCAAGTTGGGTTTGGGTTCCGTCCGA
        aaaababaaaaaaabbaaaaa[[_ab`]]Waaa^M[a\Q[     NM:i:0  NH:i:1
HANNIBAL_4_FC308YYAAXX:6:51:1238:254    0       Chr1    3766    255     40M     *       0       0       GATCAAGTTGGGTTTGGGTTCCGTCCGAACGACGAGGAGC
        aaaaaaaaaaaaaaa]X`aa[X[aa_XGHX_][X^^VU]a     NM:i:0  NH:i:1
HANNIBAL_4_FC308YYAAXX:6:85:844:367     0       Chr1    3771    255     40M     *       0       0       AGTTGGGTTTGGGTTCCGTCCGAACGACGAGGAGCTCGTT
        `_`aaaaaaUa[VXaaa_]_^QZ_URX_^^^VVXaa^[_a     NM:i:0  NH:i:1
HANNIBAL_4_FC308YYAAXX:6:19:1738:1491   0       Chr1    3834    255     40M     *       0       0       CGAAGGAAACACTAGCCGCGACGTTGAAGTAGCCATCAGT
        aa[S[WXUUUaaRE[aaaaaa[JV[[aaaaXG[ZOX[VKE     NM:i:1  NH:i:1
```

SAM (Sequence Alignment Map) Popular output format

http://samtools.sourceforge.net/

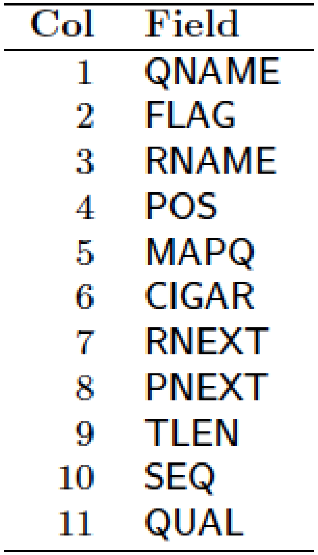
pysam - An interface for reading and writing SAM files

http://wwwfgu.anat.ox.ac.uk/~andreas/documentation/samtools/api.html



<table>
<tr><th>Col</th><th>Field</th><th>Type</th><th>Regexp/Range</th><th>Brief description</th></tr>
<tr><td>1</td><td>QNAME</td><td>String</td><td>[!-?A-~]{1,255}</td><td>Query template NAME</td></tr>
<tr><td>2</td><td>FLAG</td><td>Int</td><td>[0,2^{16}-1]</td><td>bitwise FLAG</td></tr>
<tr><td>3</td><td>RNAME</td><td>String</td><td>*|[!-()+-<>-~][!-~]*</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>*|([0-9]+[MIDNSHPX=])+</td><td>CIGAR string</td></tr>
<tr><td>7</td><td>RNEXT</td><td>String</td><td>*|=|[!-()+-<>-~][!-~]*</td><td>Ref. name of the mate/next fragment</td></tr>
<tr><td>8</td><td>PNEXT</td><td>Int</td><td>[0,2^{29}-1]</td><td>Position of the mate/next fragment</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>*|[A-Za-z=.]+</td><td>fragment SEQuence</td></tr>
<tr><td>11</td><td>QUAL</td><td>String</td><td>[!-~]+</td><td>ASCII of Phred-scaled base QUALity+33</td></tr>
</table>

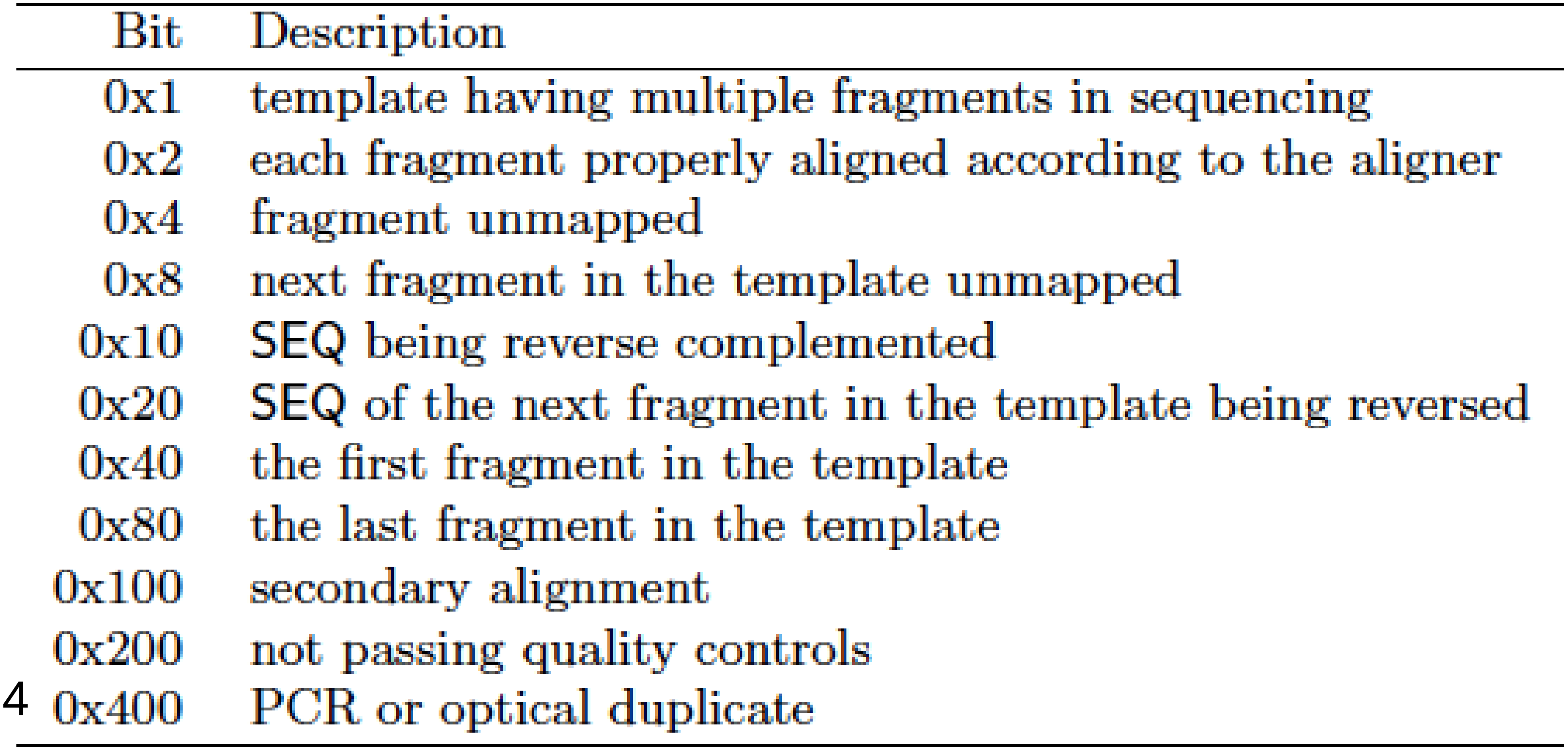
Bowtie output (SAM)



Col	Field
1	QNAME
2	FLAG
3	RNAME
4	POS
5	MAPQ
6	CIGAR
7	RNEXT
8	PNEXT
9	TLEN
10	SEQ
11	QUAL

```
1.      HYYD8:00007:00087
2.      16
3.      gb|CM000455
4.      1385117
5.      3
6.      29M1D9M1D9M2D21M2D18M1D70M
7.      *
8.      0
9.      0
10.     CAATGAGCTAACAACTGCAATGGGGCCATAATGGCTGCTTGTCGTTTGGCACGTACATGGACTAGCTTCC
        CCCGTGGCACAAAAATGGCTCTACGTTCTGTTACGAGCGCACCTACTGAAGGTCTCTCATAGGAGTGTAT
        GTATATGCATATACAT
11.     ;:=>>:333*33,33<<:7:3*344,444-449>>::4-6666B<EB>ABA@?;::44,4444<<4,4*555545-
        ??670??==?<?@?>>>><7<<45-??>>?>>>??;<44444-5,:;;<776767-
        55?667?=@@888@AA@?<>;<55
12.     AS:i:-58       XN:i:0  XM:i:4  XO:i:5  XG:i:7  NM:i:11 MD:Z:29^A9^T9^TG10C0T1G0A6^CC18^A70
        YT:Z:UU
        XR:Z:@HYYD8%3A00007%3A00087%0AATGTATATGCATATACATACACTCCTATGAGAGACCTTCAGT
        AGGTGCGCTCGTAACAGAACGTAGAGCCATTTTTGTGCCACGGGGGAAGCTAGTCCATGTACGTGCCAA
        ACGACAAGCAGCCATTATGGCCCCATTGCAGTTGTTAGCTCATTG%0A+%0A55<;><?@AA@888@@
        =?766?55-767677<;;%3A,5-44444<;??>>>?>>??-54<<7<>>>>?@?<?==??076??-
        545555*4,4<<4444,44%3A%3A;?@ABA>BE<B6666-4%3A%3A>>944-
        444,443*3%3A7%3A<<33,33*333%3A>>=%3A;%0A
```

Bitwise Flag



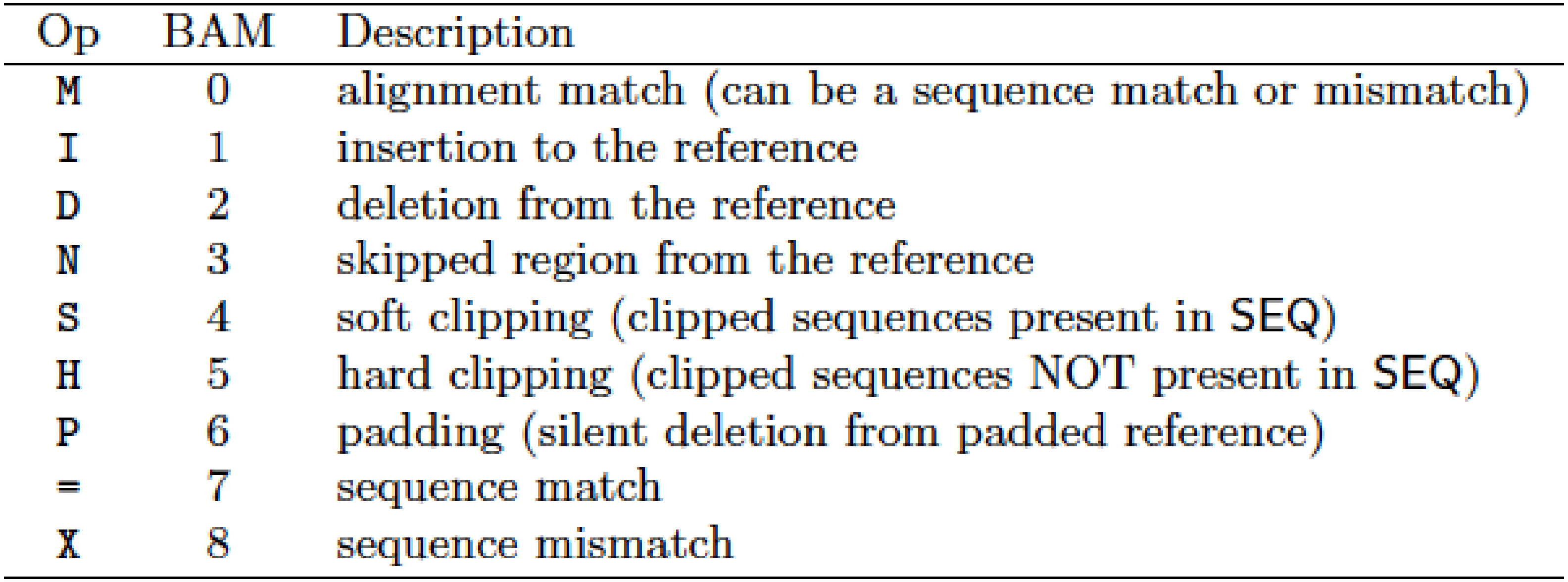
	Bit	Description
1	0x1	template having multiple fragments in sequencing
2	0x2	each fragment properly aligned according to the aligner
4	0x4	fragment unmapped
8	0x8	next fragment in the template unmapped
16	0x10	SEQ being reverse complemented
32	0x20	SEQ of the next fragment in the template being reversed
64	0x40	the first fragment in the template
128	0x80	the last fragment in the template
256	0x100	secondary alignment
512	0x200	not passing quality controls
1024	0x400	PCR or optical duplicate

What is 77? Find greatest value without going over

77-64 = 13   40
13- 8 =   5   8
5- 4 =   1   4
1- 1 =   0   1

What is 141?

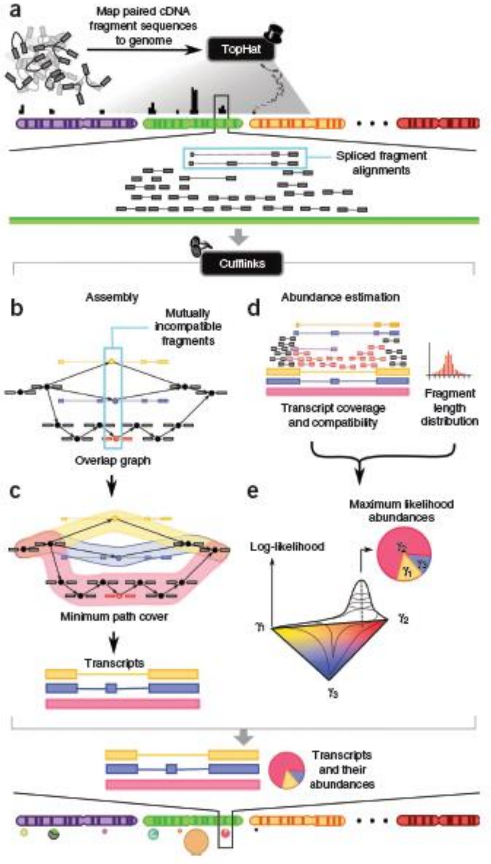
CIGAR string



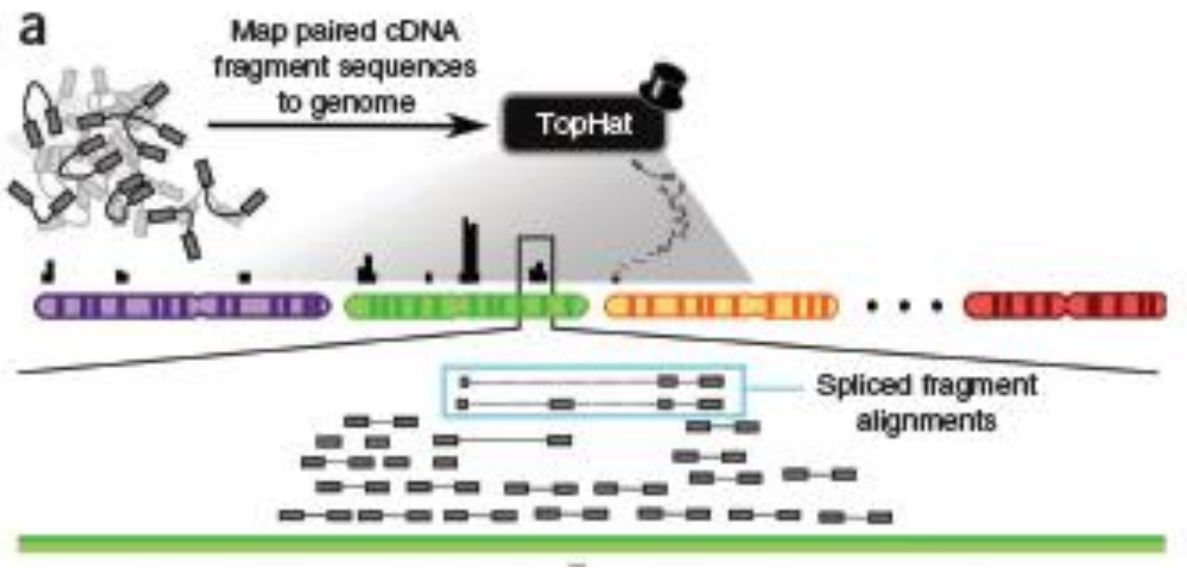
Op	BAM	Description
M	0	alignment match (can be a sequence match or mismatch)
I	1	insertion to the reference
D	2	deletion from the reference
N	3	skipped region from the reference
S	4	soft clipping (clipped sequences present in SEQ)
H	5	hard clipping (clipped sequences NOT present in SEQ)
P	6	padding (silent deletion from padded reference)
=	7	sequence match
X	8	sequence mismatch

29M 1D 9M 1D 9M 2D 21M 2D 18M 1D 70M

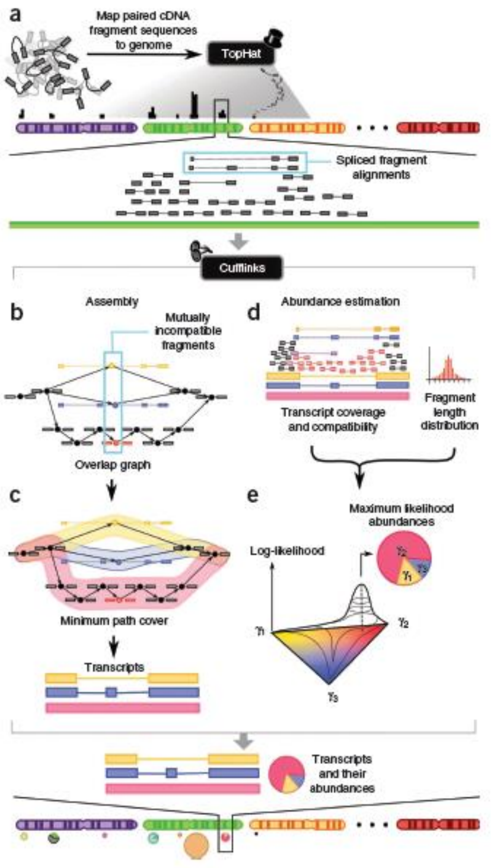
Cufflinks first starts with the output of any alignment tool such as TopHat


a
Map paired cDNA fragment sequences to genome
TopHat
Spliced fragment alignments
Cufflinks
b
Assembly
Mutually incompatible fragments
Overlap graph
c
Minimum path cover
Transcripts
d
Abundance estimation
Transcript coverage and compatibility
Fragment length distribution
e
Maximum likelihood abundances
Log-likelihood
γ_1
γ_2
γ_3
Transcripts and their abundances

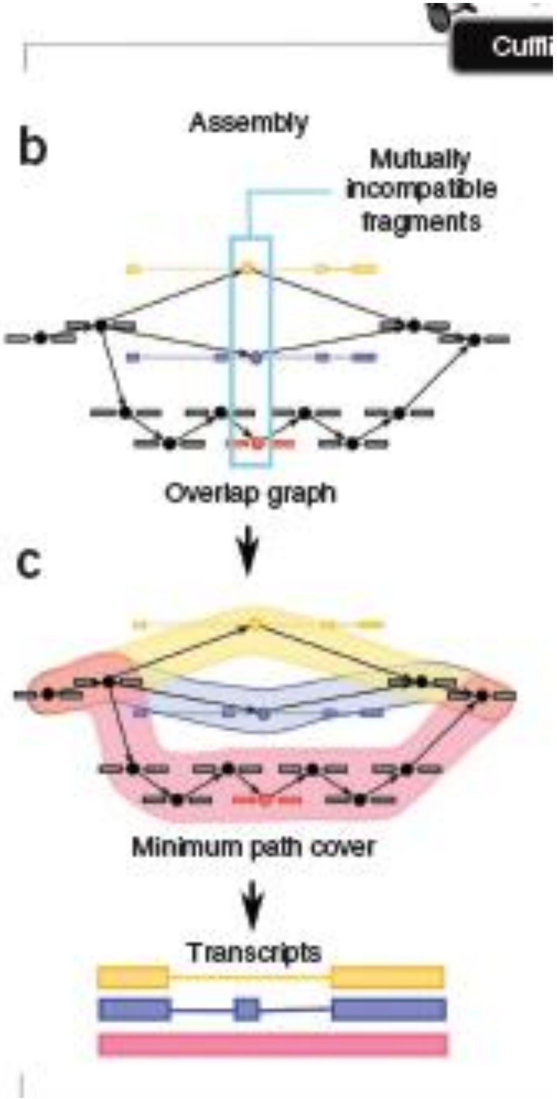
VOLUME 28 NUMBER 5 MAY 2010 NATURE BIOTECHNOLOGY


a
Map paired cDNA fragment sequences to genome
TopHat
Spliced fragment alignments

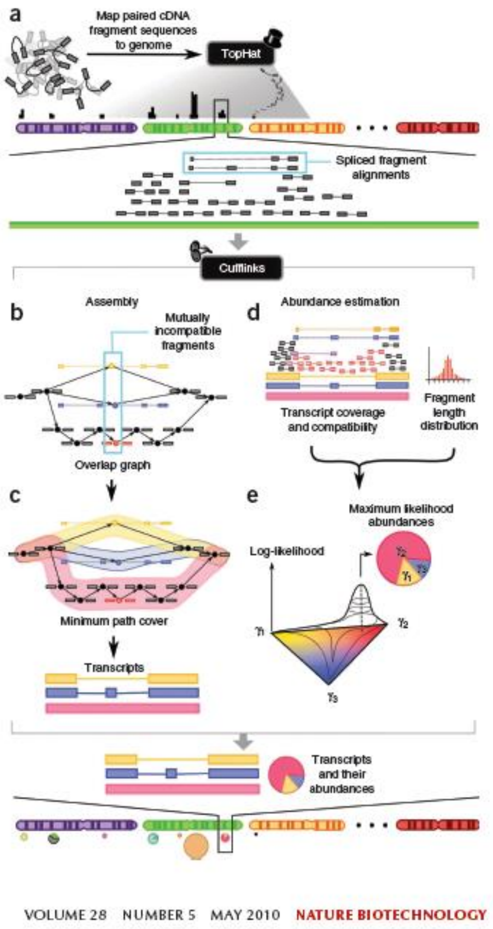
Then it assembles the isoforms by first identifying the reads that can not be assembled together.


a
Map paired cDNA fragment sequences to genome
TopHat
Spliced fragment alignments
Cufflinks
b
Assembly
Mutually incompatible fragments
Overlap graph
c
Minimum path cover
Transcripts
d
Abundance estimation
Transcript coverage and compatibility
Fragment length distribution
e
Maximum likelihood abundances
Log-likelihood
γ_1
γ_2
γ_3
Transcripts and their abundances

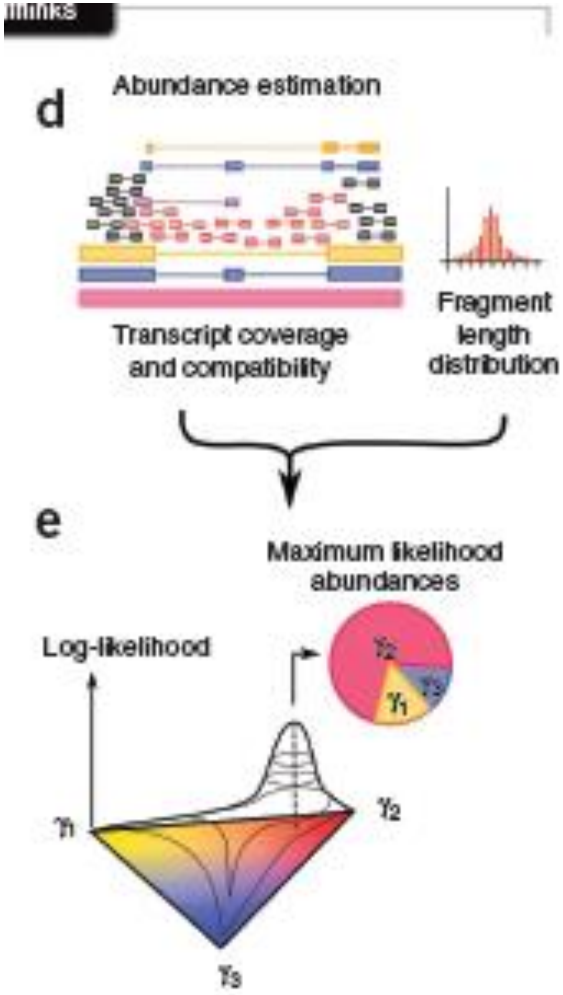
VOLUME 28 NUMBER 5 MAY 2010 NATURE BIOTECHNOLOGY



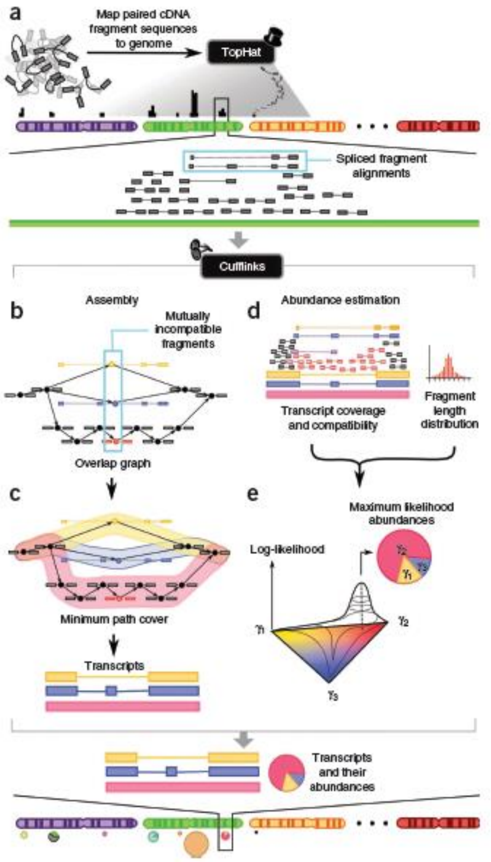
Then calculate abundance


a
Map paired cDNA fragment sequences to genome
TopHat
Spliced fragment alignments
Cufflinks
b
Assembly
Mutually incompatible fragments
Overlap graph
c
Minimum path cover
Transcripts
d
Abundance estimation
Transcript coverage and compatibility
Fragment length distribution
e
Maximum likelihood abundances
Log-likelihood
γ_1
γ_2
γ_3
Transcripts and their abundances

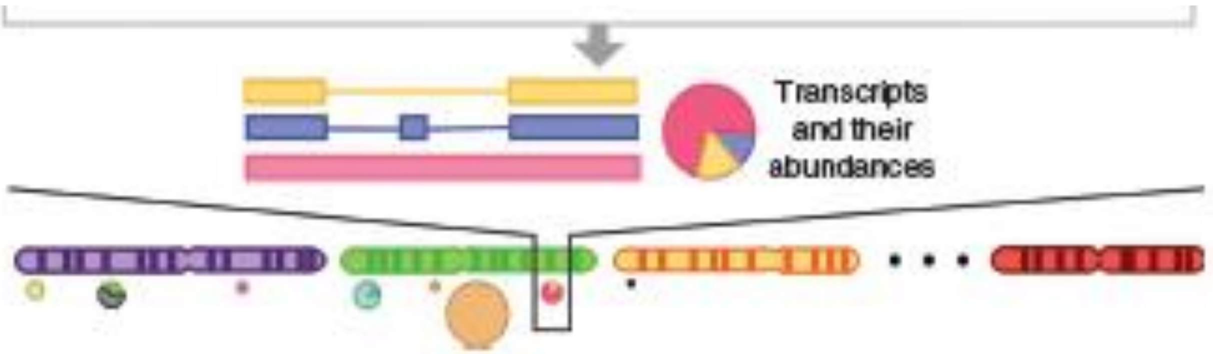
VOLUME 28 NUMBER 5 MAY 2010 NATURE BIOTECHNOLOGY


d
Abundance estimation
Transcript coverage and compatibility
Fragment length distribution
e
Maximum likelihood abundances
Log-likelihood
γ_1
γ_2
γ_3

Assembling the reads to identify transcripts.


a
Map paired cDNA fragment sequences to genome
TopHat
Spliced fragment alignments
Cufflinks
b
Assembly
Mutually incompatible fragments
Overlap graph
c
Minimum path cover
Transcripts
d
Abundance estimation
Transcript coverage and compatibility
Fragment length distribution
e
Maximum likelihood abundances
Log-likelihood
γ_1
γ_2
γ_3
Transcripts and their abundances

VOLUME 28 NUMBER 5 MAY 2010 NATURE BIOTECHNOLOGY



CuffCompare

- The program cuffcompare helps you:
  - Compare your assembled transcripts to a reference annotation
  - Track Cufflinks transcripts across multiple experiments (e.g. across a time course)
- Output contains codes
  - = match
  - c contained
  - j new isoform
  - u unknown, intergenic transcript
  - i single exon in intron region

Identification of spliced junctions depends largely on the depth of sequences coverage.

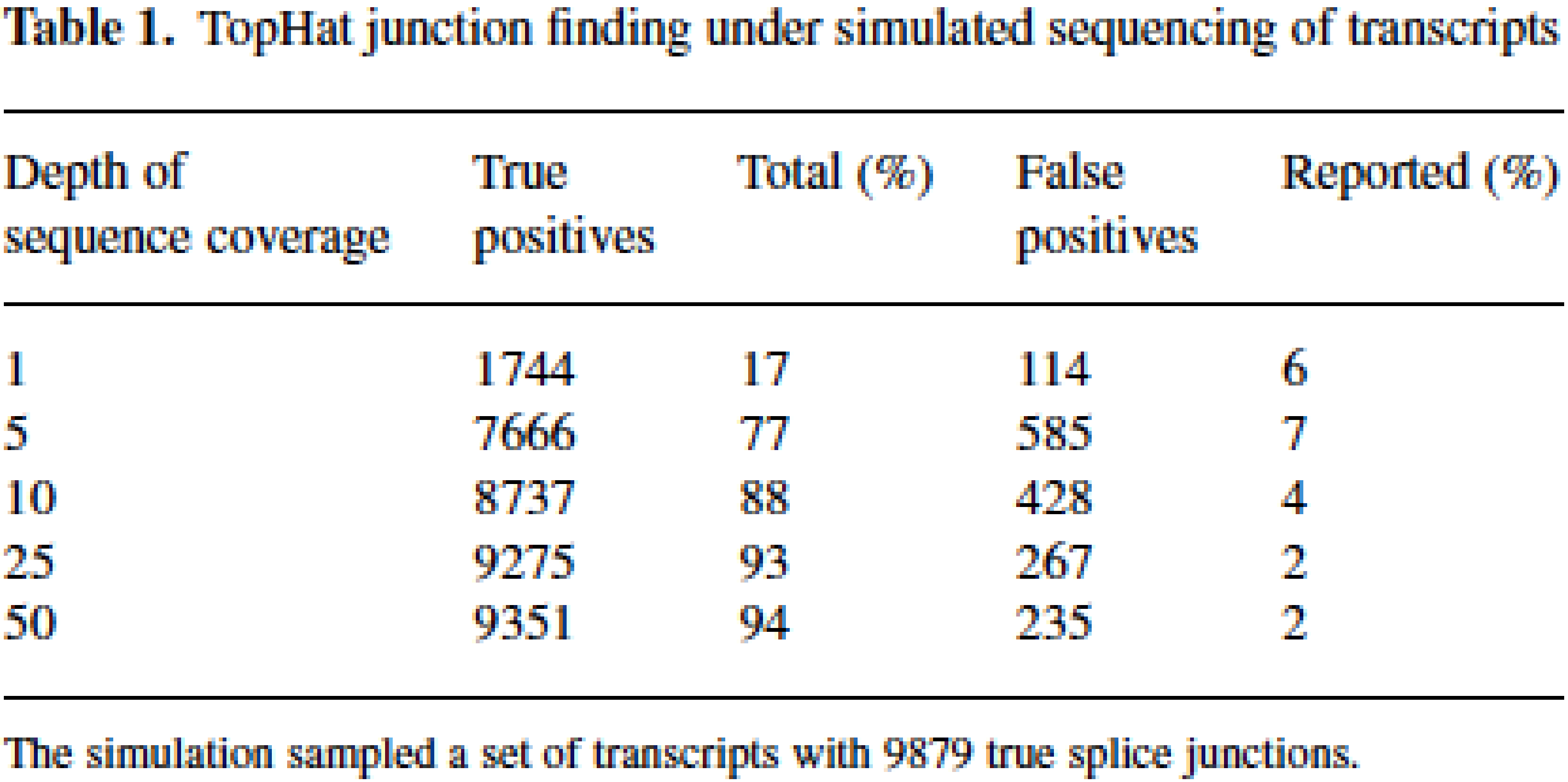


Table 1. TopHat junction finding under simulated sequencing of transcripts

Depth of sequence coverage	True positives	Total (%)	False positives	Reported (%)
1	1744	17	114	6
5	7666	77	585	7
10	8737	88	428	4
25	9275	93	267	2
50	9351	94	235	2

The simulation sampled a set of transcripts with 9879 true splice junctions.

4. Link genes and determine normalized expression values

Run cufflinks. We will enter the tophat output directory and run it in there so all cufflinks output will be in tophat_output

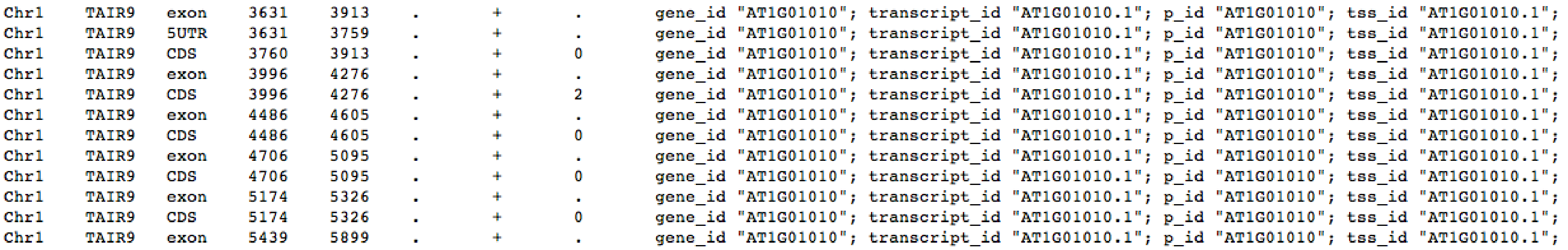
```
module load cufflinks/2.2.1
cd tophat_output
cufflinks -G /home/mkatari/manny/Arabidosis.gtf \
   accepted_hits.sam
```

```
-o = output directory
-G = GTF file
sam output file
```

GTF files are like GFF file but with specific attributes

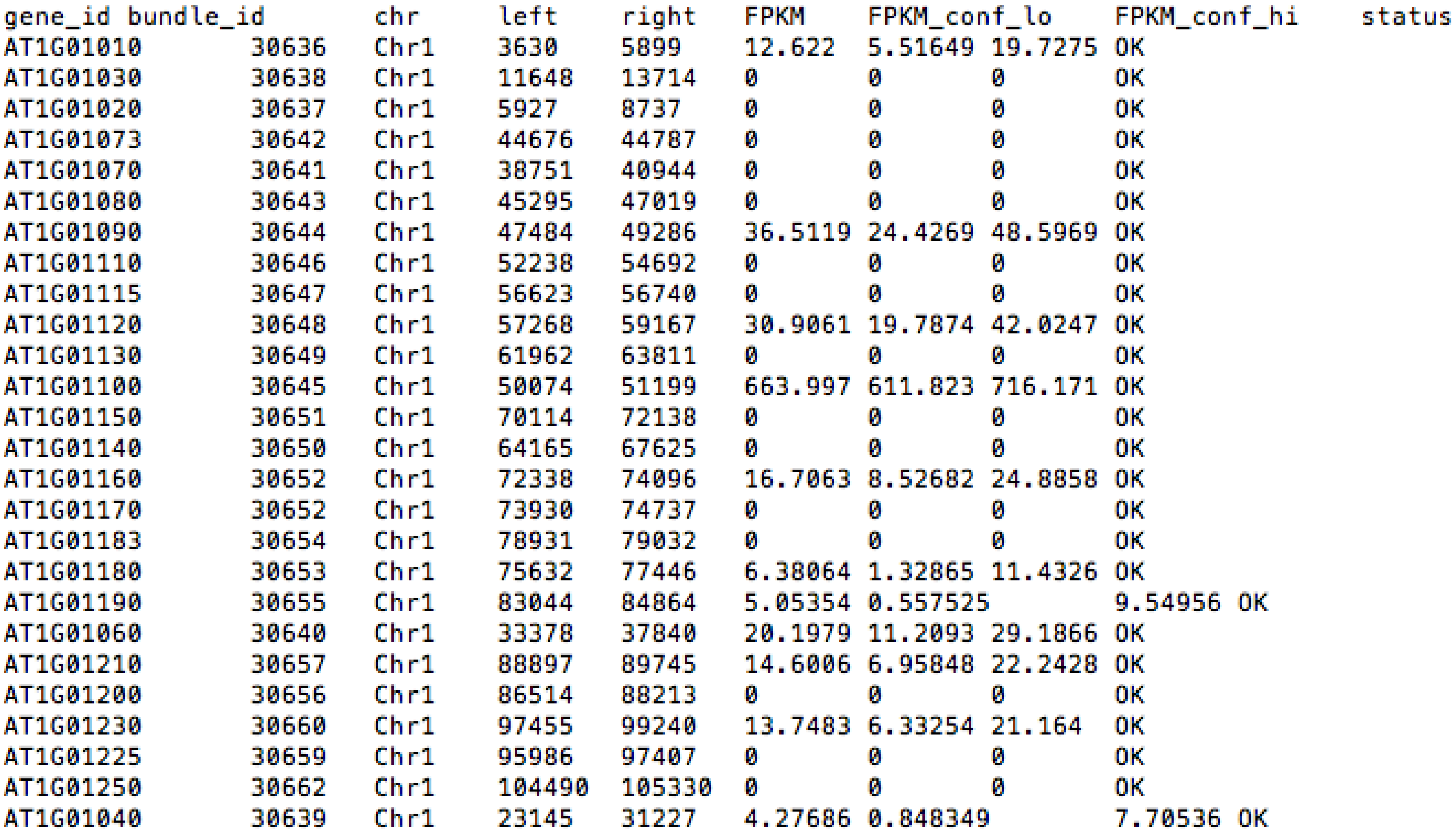
- If you find a gff file for your organism, you can easily convert it into a gtf file. TAIR10_GFF3_genes.gff is a file I downloaded from TAIR which contains coordinates of the genes annotated in Arabidopsis.

```
gffread -E TAIR10_GFF3_genes.gff -T -o- > Arabidopsis.gtf
```



```
Chr1	TAIR9	exon	3631	3913	.	+	.	gene_id "AT1G01010"; transcript_id "AT1G01010.1"; p_id "AT1G01010"; tss_id "AT1G01010.1";
Chr1	TAIR9	5UTR	3631	3759	.	+	.	gene_id "AT1G01010"; transcript_id "AT1G01010.1"; p_id "AT1G01010"; tss_id "AT1G01010.1";
Chr1	TAIR9	CDS	3760	3913	.	+	0	gene_id "AT1G01010"; transcript_id "AT1G01010.1"; p_id "AT1G01010"; tss_id "AT1G01010.1";
Chr1	TAIR9	exon	3996	4276	.	+	.	gene_id "AT1G01010"; transcript_id "AT1G01010.1"; p_id "AT1G01010"; tss_id "AT1G01010.1";
Chr1	TAIR9	CDS	3996	4276	.	+	2	gene_id "AT1G01010"; transcript_id "AT1G01010.1"; p_id "AT1G01010"; tss_id "AT1G01010.1";
Chr1	TAIR9	exon	4486	4605	.	+	.	gene_id "AT1G01010"; transcript_id "AT1G01010.1"; p_id "AT1G01010"; tss_id "AT1G01010.1";
Chr1	TAIR9	CDS	4486	4605	.	+	0	gene_id "AT1G01010"; transcript_id "AT1G01010.1"; p_id "AT1G01010"; tss_id "AT1G01010.1";
Chr1	TAIR9	exon	4706	5095	.	+	.	gene_id "AT1G01010"; transcript_id "AT1G01010.1"; p_id "AT1G01010"; tss_id "AT1G01010.1";
Chr1	TAIR9	CDS	4706	5095	.	+	0	gene_id "AT1G01010"; transcript_id "AT1G01010.1"; p_id "AT1G01010"; tss_id "AT1G01010.1";
Chr1	TAIR9	exon	5174	5326	.	+	.	gene_id "AT1G01010"; transcript_id "AT1G01010.1"; p_id "AT1G01010"; tss_id "AT1G01010.1";
Chr1	TAIR9	CDS	5174	5326	.	+	0	gene_id "AT1G01010"; transcript_id "AT1G01010.1"; p_id "AT1G01010"; tss_id "AT1G01010.1";
Chr1	TAIR9	exon	5439	5899	.	+	.	gene_id "AT1G01010"; transcript_id "AT1G01010.1"; p_id "AT1G01010"; tss_id "AT1G01010.1";
```

Cufflinks output: fpkm normalized values



```
gene_id bundle_id       chr     left    right   FPKM    FPKM_conf_lo    FPKM_conf_hi    status
AT1G01010       30636   Chr1    3630    5899    12.622  5.51649 19.7275 OK
AT1G01030       30638   Chr1    11648   13714   0       0       0       OK
AT1G01020       30637   Chr1    5927    8737    0       0       0       OK
AT1G01073       30642   Chr1    44676   44787   0       0       0       OK
AT1G01070       30641   Chr1    38751   40944   0       0       0       OK
AT1G01080       30643   Chr1    45295   47019   0       0       0       OK
AT1G01090       30644   Chr1    47484   49286   36.5119 24.4269 48.5969 OK
AT1G01110       30646   Chr1    52238   54692   0       0       0       OK
AT1G01115       30647   Chr1    56623   56740   0       0       0       OK
AT1G01120       30648   Chr1    57268   59167   30.9061 19.7874 42.0247 OK
AT1G01130       30649   Chr1    61962   63811   0       0       0       OK
AT1G01100       30645   Chr1    50074   51199   663.997 611.823 716.171 OK
AT1G01150       30651   Chr1    70114   72138   0       0       0       OK
AT1G01140       30650   Chr1    64165   67625   0       0       0       OK
AT1G01160       30652   Chr1    72338   74096   16.7063 8.52682 24.8858 OK
AT1G01170       30652   Chr1    73930   74737   0       0       0       OK
AT1G01183       30654   Chr1    78931   79032   0       0       0       OK
AT1G01180       30653   Chr1    75632   77446   6.38064 1.32865 11.4326 OK
AT1G01190       30655   Chr1    83044   84864   5.05354 0.557525        9.54956 OK
AT1G01060       30640   Chr1    33378   37840   20.1979 11.2093 29.1866 OK
AT1G01210       30657   Chr1    88897   89745   14.6006 6.95848 22.2428 OK
AT1G01200       30656   Chr1    86514   88213   0       0       0       OK
AT1G01230       30660   Chr1    97455   99240   13.7483 6.33254 21.164  OK
AT1G01225       30659   Chr1    95986   97407   0       0       0       OK
AT1G01250       30662   Chr1    104490  105330  0       0       0       OK
AT1G01040       30639   Chr1    23145   31227   4.27686 0.848349        7.70536 OK
```

Cuffdiff

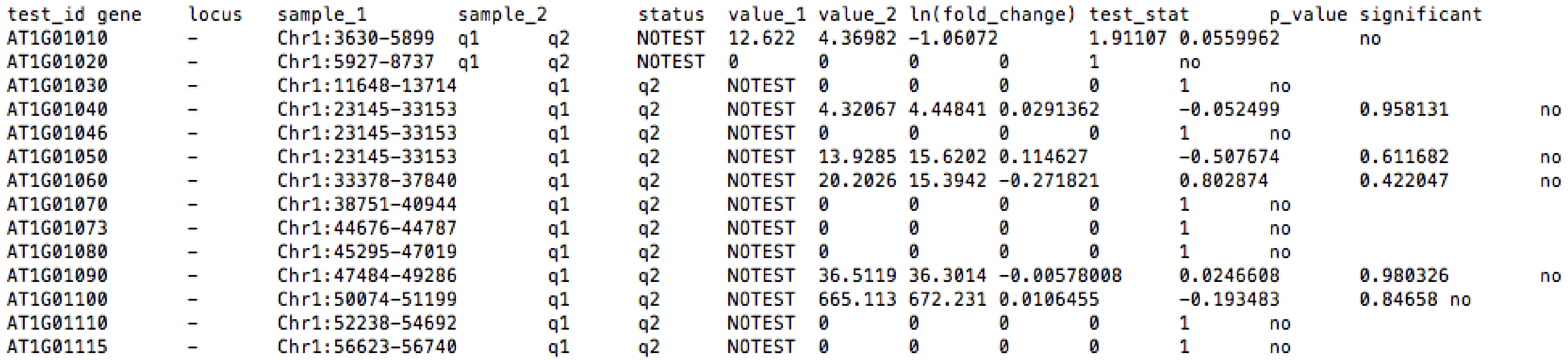
- Can be use to find significant changes in transcript expression, splicing, and promoter use.
  - Inputs are:
    - Annotation to compare (can be output from cufflinks)
    - Tophat output from different samples
    - Cuffdiff allows to compare samples even if you have only replicate.

5. Compare expression values of two sample

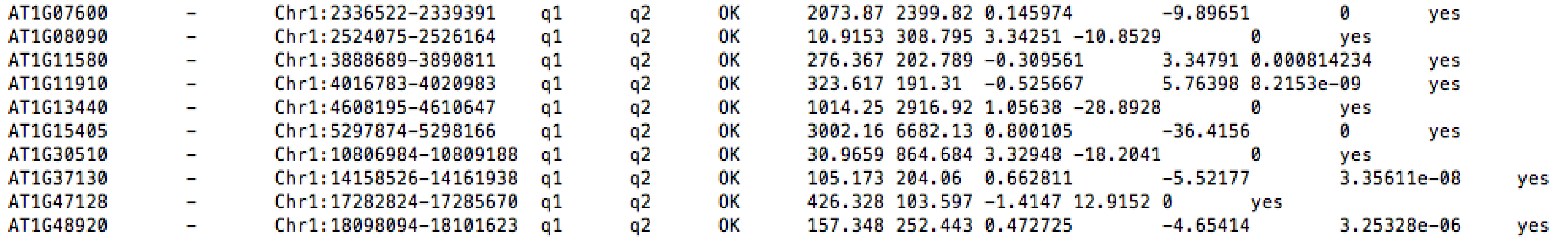
Run cuffdiff - data is normalized and a modified version of t-test is used and p-values are corrected for multiple hypothesis testing.

```
cuffdiff -o cuff_diff \
   -L KCL,NO3 \
   --dispersion-method poisson \
   --library-norm-method quartile \
   /home/mkatari/nitrogen/Arabidopsis.gtf \
   /home/mkatari/nitrogen/KCL1/accepted_hits.bam,/home/mkatari/nitrogen/KCL2/accepted_hits.bam \
   /home/mkatari/nitrogen/NO31/accepted_hits.sam,/home/mkatari/nitrogen/NO32/accepted_hits.bam
```

Cuffdiff results



```
test_id gene    locus   sample_1        sample_2        status  value_1 value_2 ln(fold_change) test_stat       p_value significant
AT1G01010       -       Chr1:3630-5899  q1      q2      NOTEST  12.622  4.36982 -1.06072        1.91107 0.0559962       no
AT1G01020       -       Chr1:5927-8737  q1      q2      NOTEST  0       0       0       0       1       no
AT1G01030       -       Chr1:11648-13714        q1      q2      NOTEST  0       0       0       0       1       no
AT1G01040       -       Chr1:23145-33153        q1      q2      NOTEST  4.32067 4.44841 0.0291362       -0.052499       0.958131        no
AT1G01046       -       Chr1:23145-33153        q1      q2      NOTEST  0       0       0       0       1       no
AT1G01050       -       Chr1:23145-33153        q1      q2      NOTEST  13.9285 15.6202 0.114627        -0.507674       0.611682        no
AT1G01060       -       Chr1:33378-37840        q1      q2      NOTEST  20.2026 15.3942 -0.271821       0.802874        0.422047        no
AT1G01070       -       Chr1:38751-40944        q1      q2      NOTEST  0       0       0       0       1       no
AT1G01073       -       Chr1:44676-44787        q1      q2      NOTEST  0       0       0       0       1       no
AT1G01080       -       Chr1:45295-47019        q1      q2      NOTEST  0       0       0       0       1       no
AT1G01090       -       Chr1:47484-49286        q1      q2      NOTEST  36.5119 36.3014 -0.00578008     0.0246608       0.980326        no
AT1G01100       -       Chr1:50074-51199        q1      q2      NOTEST  665.113 672.231 0.0106455       -0.193483       0.84658 no
AT1G01110       -       Chr1:52238-54692        q1      q2      NOTEST  0       0       0       0       1       no
AT1G01115       -       Chr1:56623-56740        q1      q2      NOTEST  0       0       0       0       1       no
```

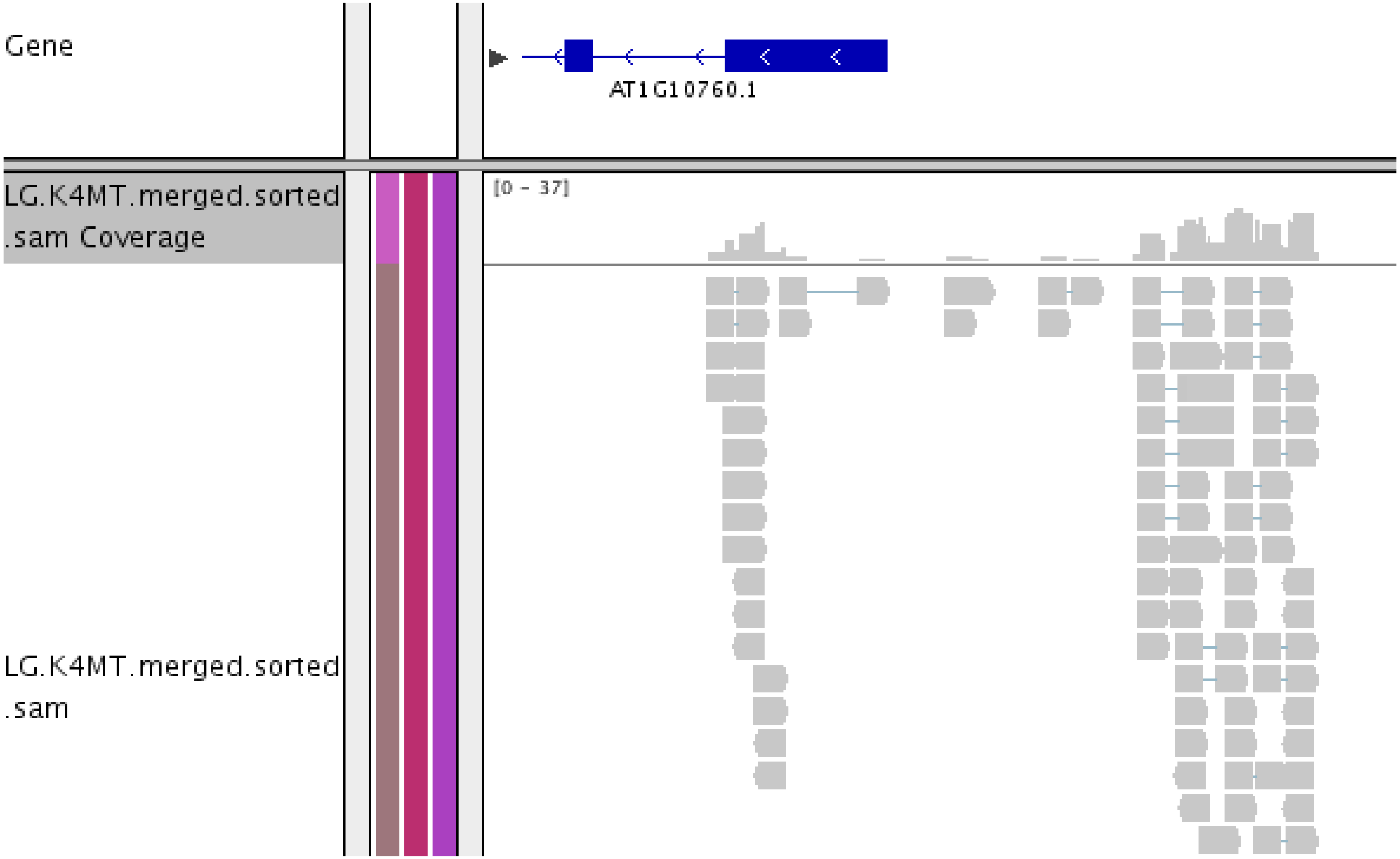


```
AT1G07600       -       Chr1:2336522-2339391    q1      q2      OK      2073.87 2399.82 0.145974        -9.89651        0       yes
AT1G08090       -       Chr1:2524075-2526164    q1      q2      OK      10.9153 308.795 3.34251 -10.8529        0       yes
AT1G11580       -       Chr1:3888689-3890811    q1      q2      OK      276.367 202.789 -0.309561       3.34791 0.000814234     yes
AT1G11910       -       Chr1:4016783-4020983    q1      q2      OK      323.617 191.31  -0.525667       5.76398 8.2153e-09      yes
AT1G13440       -       Chr1:4608195-4610647    q1      q2      OK      1014.25 2916.92 1.05638 -28.8928        0       yes
AT1G15405       -       Chr1:5297874-5298166    q1      q2      OK      3002.16 6682.13 0.800105        -36.4156        0       yes
AT1G30510       -       Chr1:10806984-10809188  q1      q2      OK      30.9659 864.684 3.32948 -18.2041        0       yes
AT1G37130       -       Chr1:14158526-14161938  q1      q2      OK      105.173 204.06  0.662811        -5.52177        3.35611e-08     yes
AT1G47128       -       Chr1:17282824-17285670  q1      q2      OK      426.328 103.597 -1.4147 12.9152 0       yes
AT1G48920       -       Chr1:18098094-18101623  q1      q2      OK      157.348 252.443 0.472725        -4.65414        3.25328e-06     yes
```

To get a list of genes that are significantly differentially expressed genes

```
cut -f 1,8,9,10,12,13,14 gene_exp.diff | grep "yes" | less
```

IGV



Gene
AT1G10760.1
LG.K4MT.merged.sorted.sam Coverage
[0 - 37]
LG.K4MT.merged.sorted.sam

IGV: Integrative Genomics Viewer

- http://www.broadinstitute.org/igv/
- Standalone java program
  - Does not require a mysql database server or an apache web server
  - Limited to the resources of the machine that it is running on.
  - More interactive compared to Gbrowse.
  - Both IGV and Gbrowse can use GFF file format.

IGV tools

- http://www.broadinstitute.org/igv/igvtools
- Simple tools to format the files so you can use them on the browser
- Tools that I have needed:
  - Sort
  - Index

Visualizing in IGV

- There are two main steps:
  - You have to index the reference sequence
  - Sort and index the bam files

```
module load samtools/0.1.19
samtools faidx Arabidopsis.fa

samtools sort KCL1/accepted_hits.bam KCL1_sorted.bam
samtools index KCL1_sorted.bam
samtools sort KCL2/accepted_hits.bam KCL2_sorted.bam
samtools index KCL1_sorted.bam
samtools sort NO31/accepted_hits.sam NO31_sorted.bam
samtools index KCL1_sorted.bam
samtools sort NO32/accepted_hits.bam NO32_sorted.bam
samtools index KCL1_sorted.bam
```

Launch IGV

```
module load module load igv/2.1.21
igv

First load the genome fasta sequence
File->Load Genome from File
   Select the Arabidopsis.fa in /home/mkatari/nitrogen/

Then load the annotations
File->Load from File
   Select the Arabidopsis.gft in /home/mkatari/nitrogen/

Then load as many alignments you would like
File->Load from File
   Select the KCL1_sorted.bam in /home/mkatari/nitrogen/

IGV is also capable of visualizing VCF(Variant call format)
```

Don’t forget to Dedup !!

- Sequencing platforms are not perfect. Occasionally one dna fragment can generate many clusters. You know you have this problem when the reads are exactly the same and are generated using random sequencing.

```
module load samtools/0.1.19

samtools sort KCL1/accepted_hits.bam
   KCL1/accepted_hits_sorted

samtools rmdup -s KCL1/accepted_hits_sorted.bam
   KCL1/accepted_dedup.bam
```

Assignment

- Repeat the Cuffdiff step using dedup alignment files
  - How many genes are differentially expressed now ?
- Repeat the sorting and indexing for the deduped files so we can visualize them on IGV.
  - Make sure the artifacts have all gone.
  - Confirm results by looking at AT1G77760
  - Is it higher or lower in the presence of Nitrate?