RNA-seq analysis

Quantification

Alexey Sergushichev

2021-08-26, Tomsk

Overview of the day

- ✓ **RNA-seq quantification** – from raw data to an expression table
- ✓ (RNA-seq analysis in R – from an expression table to pathway analysis)
- ✓ Visual gene expression analysis in Phantasus

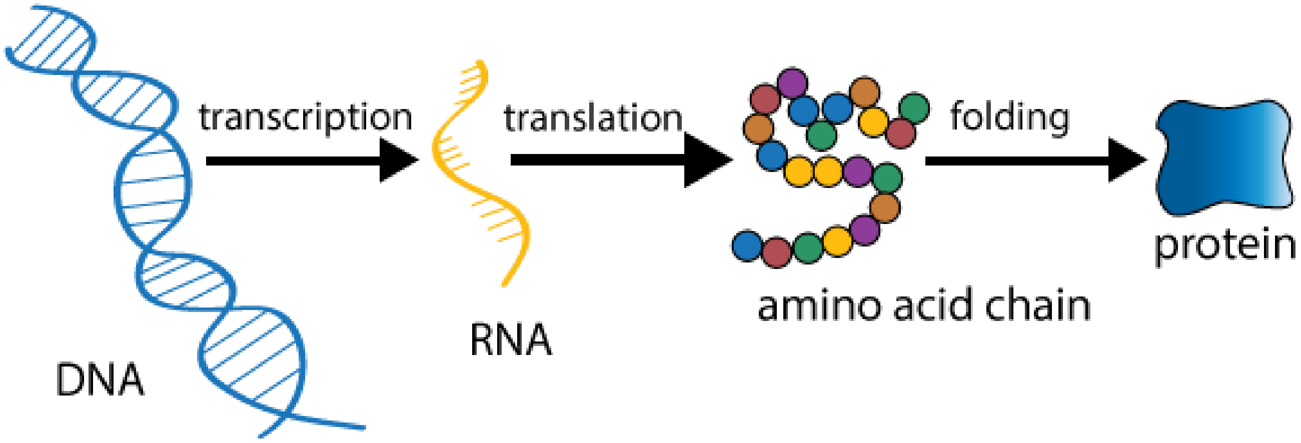
- ✓ Materials and slides are available at Google Drive
- ✓ Dockerfile and the scripts are available at https://github.com/ctlab/sysbio-training/tree/master/tomsk-scs-2021/

Prepare

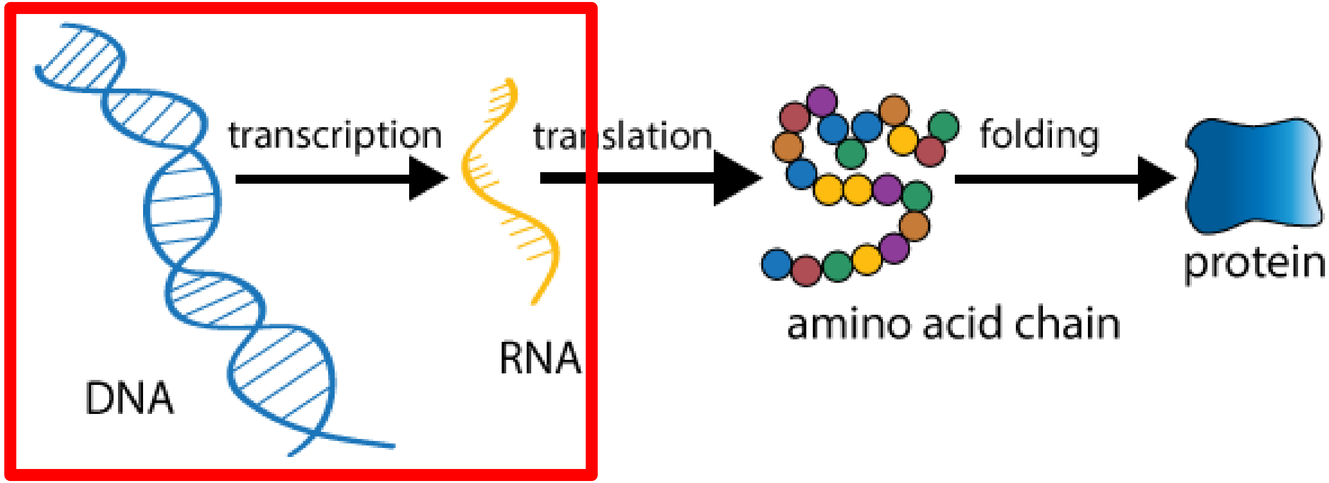
- ✓ Go to https://ctlab.itmo.ru/rstudio-sbNN/
- ✓ login: student
- ✓ password: sysbiopass

- ✓ Open project “rnaseq”
  - File -> Open project -> rnaseq -> rnaseq.Rproj -> Open
- ✓ Look around a bit

Gene expression = transcription

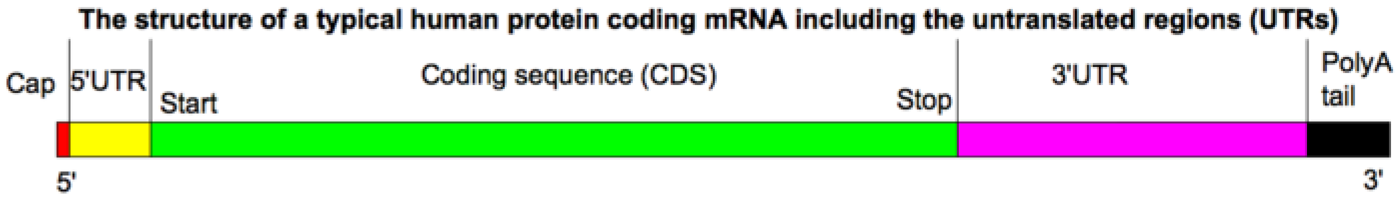


Study transcription (because we can)



mRNA

- Protein-coding RNA
- Estimated 0 – 300000 copies of each gene’s mRNA per animal cell
- mRNA levels correlate with protein levels


The structure of a typical human protein coding mRNA including the untranslated regions (UTRs)
Cap
5'UTR
Start
Coding sequence (CDS)
Stop
3'UTR
PolyA
tail
5'
3'

Main types of RNAs

- rRNA – ribosomal RNA: 80% of the cell RNA
- tRNA – transfer RNA: 15% of the cell RNA
- **mRNA** – messenger RNA for protein coding genes
- Other RNAs: miRNA, lncRNA, …

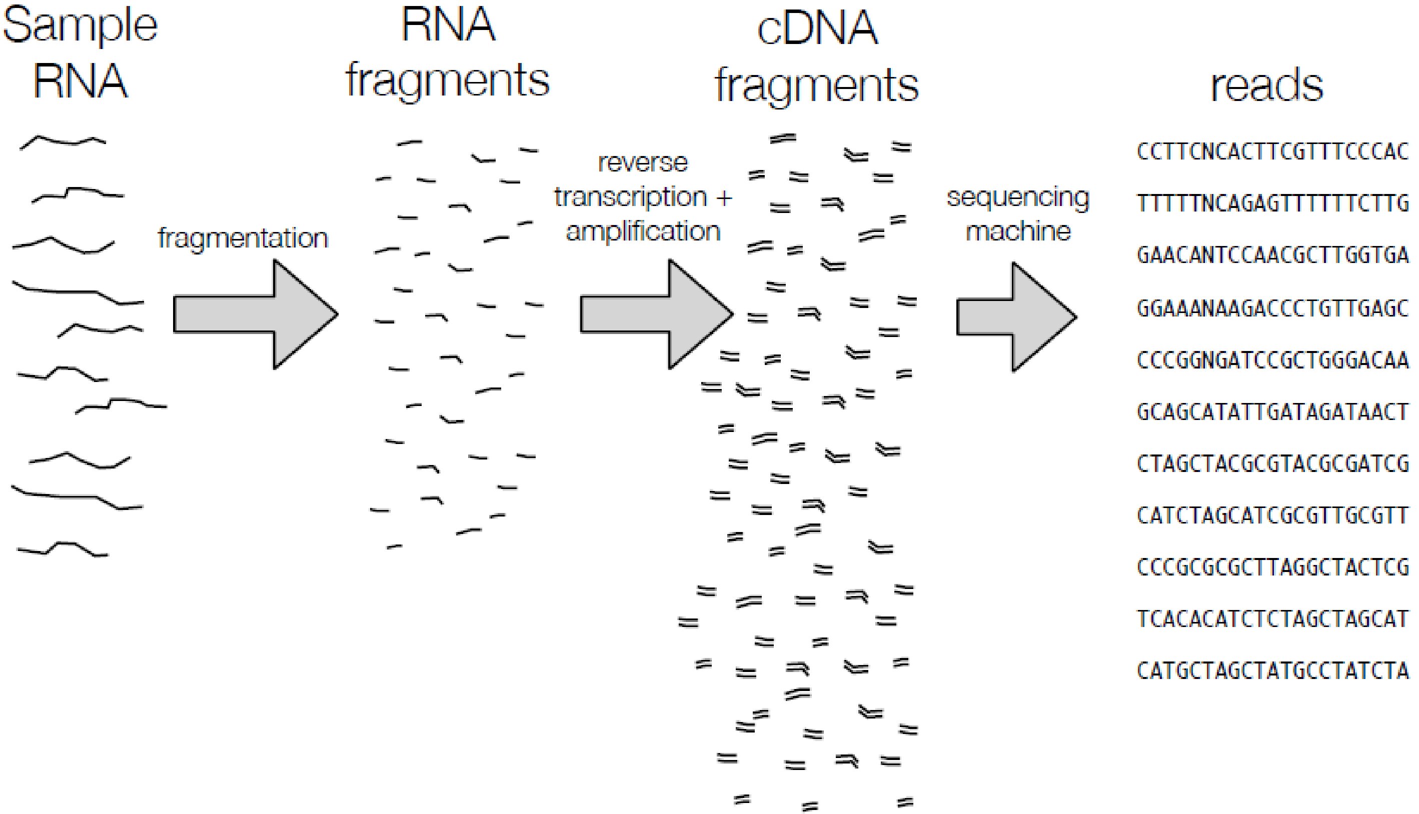
- Some of the RNAs are short: tRNA, miRNA, … and are not getting into normal RNA-seq

https://www.ncbi.nlm.nih.gov/books/NBK21729/

Two main approaches for RNA selection

- ✓ polyA selection: most standard, relatively cheap and easy protocol, selects mRNAs and some non-coding RNAs
- ✓ riboZero: depletes rRNA, works better for degraded RNA, captures all long RNAs

What is RNA-seq



```
CCTTCNCACTTCGTTTCCCAC
TTTTTNCAGAGTTTTTTCTTG
GAACANTCCAACGCTTGGTGA
GGAAANAAGACCCTGTTGAGC
CCCGGNGATCCGCTGGGACAA
GCAGCATATTGATAGATAACT
CTAGCTACGCGTACGCGATCG
CATCTAGCATCGCGTTGCGTT
CCCGCGCGCTTAGGCTACTCG
TCACACATCTCTAGCTAGCAT
CATGCTAGCTATGCCTATCTA
```

http://www.biostat.wisc.edu/bmi776/lectures/rnaseq.pdf

Two distinct types of RNA-seq

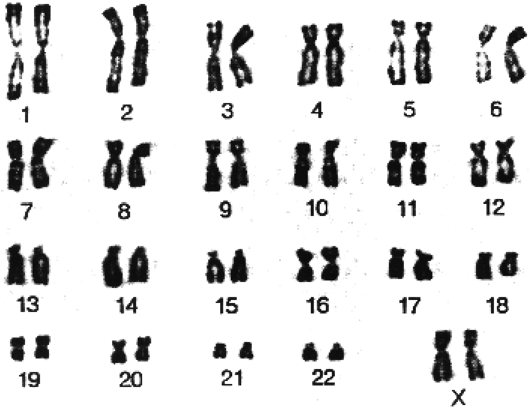
- model organism with good reference genome
- non-model organism with no/poor reference genome

Well studied

Not so much

Well-defined genomes

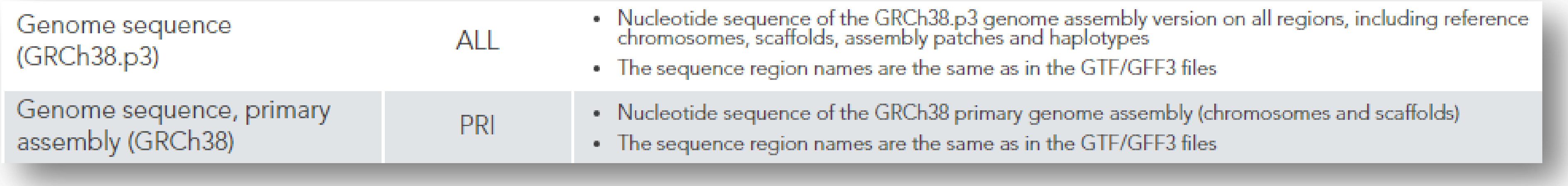
- ✓ Human: chr1-22, chrX, chrY, chrM,
  - 3235 Mb, 19815 genes
- ✓ Mouse: chr1-19, chrX, chrY, chrM,
  - 2718 Mb, 21971 genes
- ✓ Assembly is mostly complete, but not 100% - there are unplaced scaffolds and gaps
- ✓ there are rRNAs and few genes in the patches, which could be important



http://www.slideshare.net/hhalhaddad/the-human-genome-project-part-iii

Popular genome assemblies

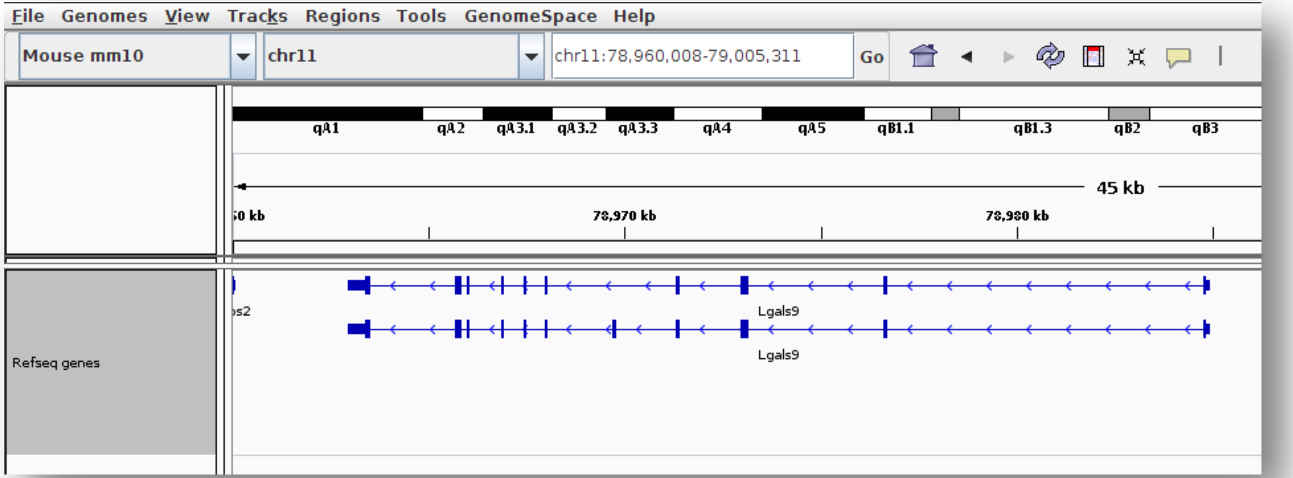
- ✓ Human:
  - UCSC notation (hg19, hg38)
  - Genome reference consortium notation (major: GRCh37, minor: GRCh38.p7)
  - 1000 genomes notation (b37)
- ✓ Mouse – same (mm10, GRCm37)



Genome sequence (GRCh38.p3)	ALL	• Nucleotide sequence of the GRCh38.p3 genome assembly version on all regions, including reference chromosomes, scaffolds, assembly patches and haplotypes • The sequence region names are the same as in the GTF/GFF3 files
Genome sequence, primary assembly (GRCh38)	PRI	• Nucleotide sequence of the GRCh38 primary genome assembly (chromosomes and scaffolds) • The sequence region names are the same as in the GTF/GFF3 files

What’s a gene?

- DNA is transcribed a lot, giving multiple types of RNA
- Some encode proteins, some do not
- Set of transcripts with a similar function = gene
- For a canonical protein-coding gene, transcripts = isoforms

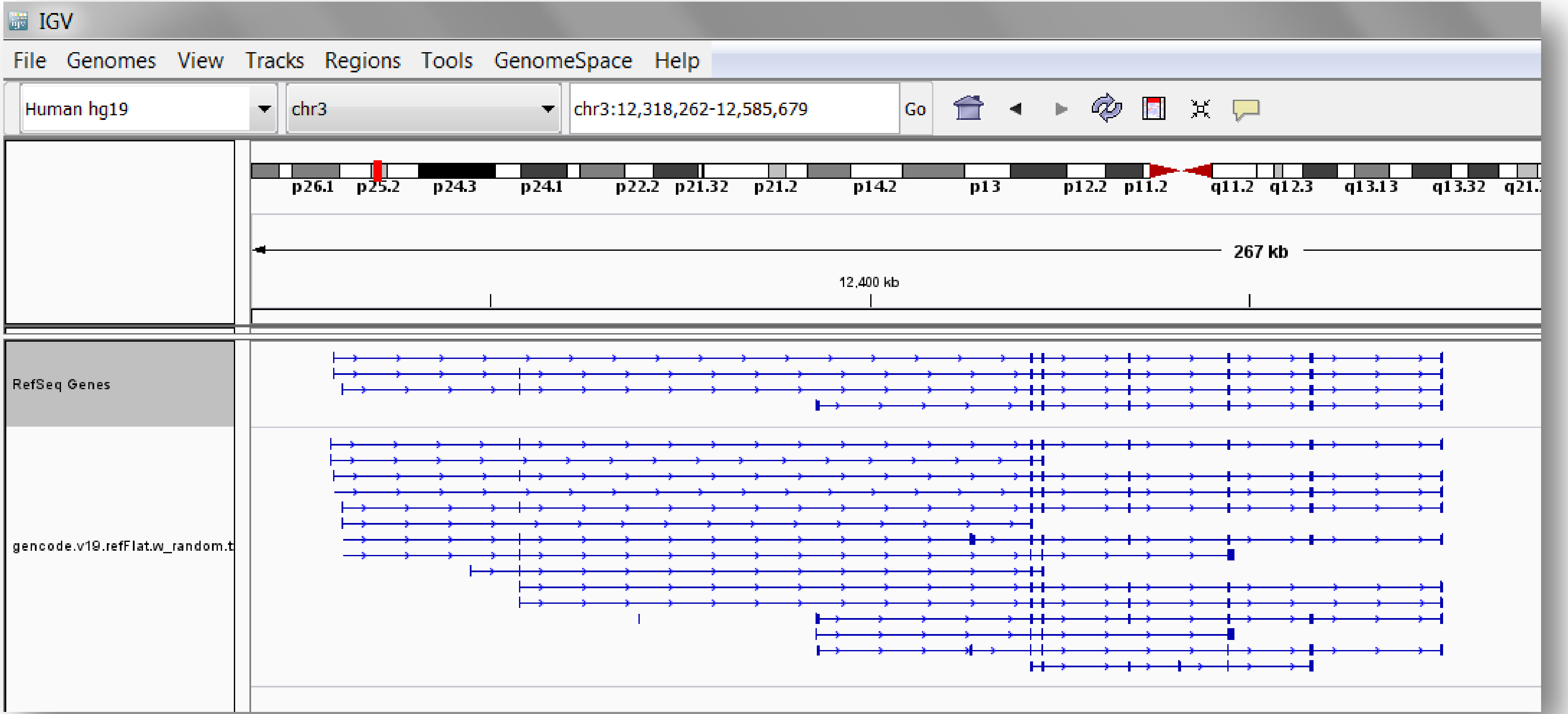


Mouse mm10

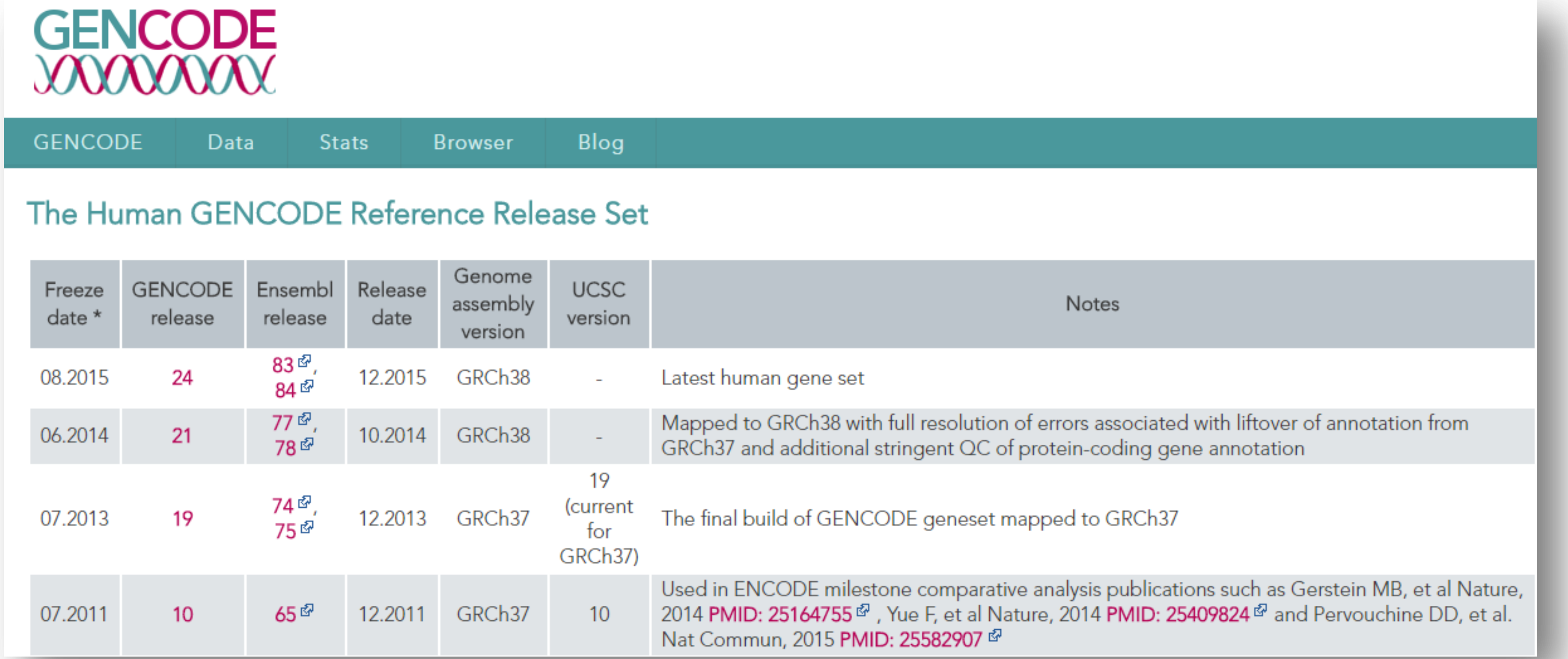
chr11

chr11:78,960,008-79,005,311

Annotation matters: RefSeq, ENSEMBL, Gencode



Just use Gencode, if you can



The Human GENCODE Reference Release Set

Freeze date *	GENCODE release	Ensembl release	Release date	Genome assembly version	UCSC version	Notes
08.2015	24	83, 84	12.2015	GRCh38	-	Latest human gene set
06.2014	21	77, 78	10.2014	GRCh38	-	Mapped to GRCh38 with full resolution of errors associated with liftover of annotation from GRCh37 and additional stringent QC of protein-coding gene annotation
07.2013	19	74, 75	12.2013	GRCh37	19 (current for GRCh37)	The final build of GENCODE geneset mapped to GRCh37
07.2011	10	65	12.2011	GRCh37	10	Used in ENCODE milestone comparative analysis publications such as Gerstein MB, et al Nature, 2014 PMID: 25164755 , Yue F, et al Nature, 2014 PMID: 25409824 and Pervouchine DD, et al. Nat Commun, 2015 PMID: 25582907

https://www.gencodegenes.org/mouse/

Reference genome

- Open ~/shared/RNAseq/reference/Gencode_mouse/release_M20/ in file browser
- In terminal:
  - `cd ~/shared/RNAseq/reference/Gencode_mouse/release_M20/`
  - `head GRCm38.primary_assembly.genome.fa`
  - `tail GRCm38.primary_assembly.genome.fa`
- get_genome.sh – scripts to download reference files

Raw sequence file formats: FASTA (.fa, .fasta)

- Very simple format (used e.g. for reference genomes)
- Has two lines:
  - sequence name (starts with “>”)
  - sequence

```
>read123456
NGGGCCAAAGGAGCTTTCAAGGAGAGAAAGAGAAGAAATAGAGAAGCAAA
```

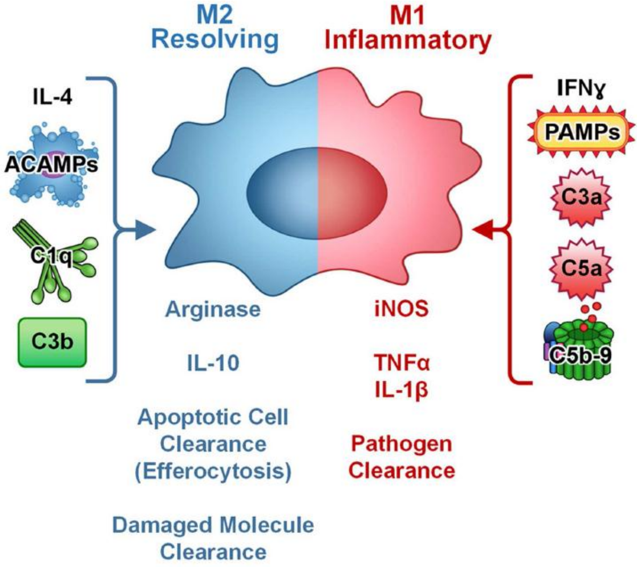
Reference annotation

✓ In terminal:

- `head gencode.vM20.annotation.gtf`
- `zcat gencode.vM20.transcripts.fa.gz | head`

Our dataset: M1 murine macrophages

- M1 (pro-inflammatory) phenotype is achieved by LPS treatment
- GSE120762



http://dx.doi.org/10.3389/fimmu.2014.00402

Go to GEO

- ✓ https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE120762
- ✓ https://www.ncbi.nlm.nih.gov/sra?term=SRP163147
  - Go to “Run selector”
- ✓ https://www.ncbi.nlm.nih.gov//bioproject/PRJNA494404
- ✓ https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?study=SRP163147

- ✓ `fastq-dump` program can be used to download the files
  - faster with `prefetch` and `parallel-fastq-dump`

Input data

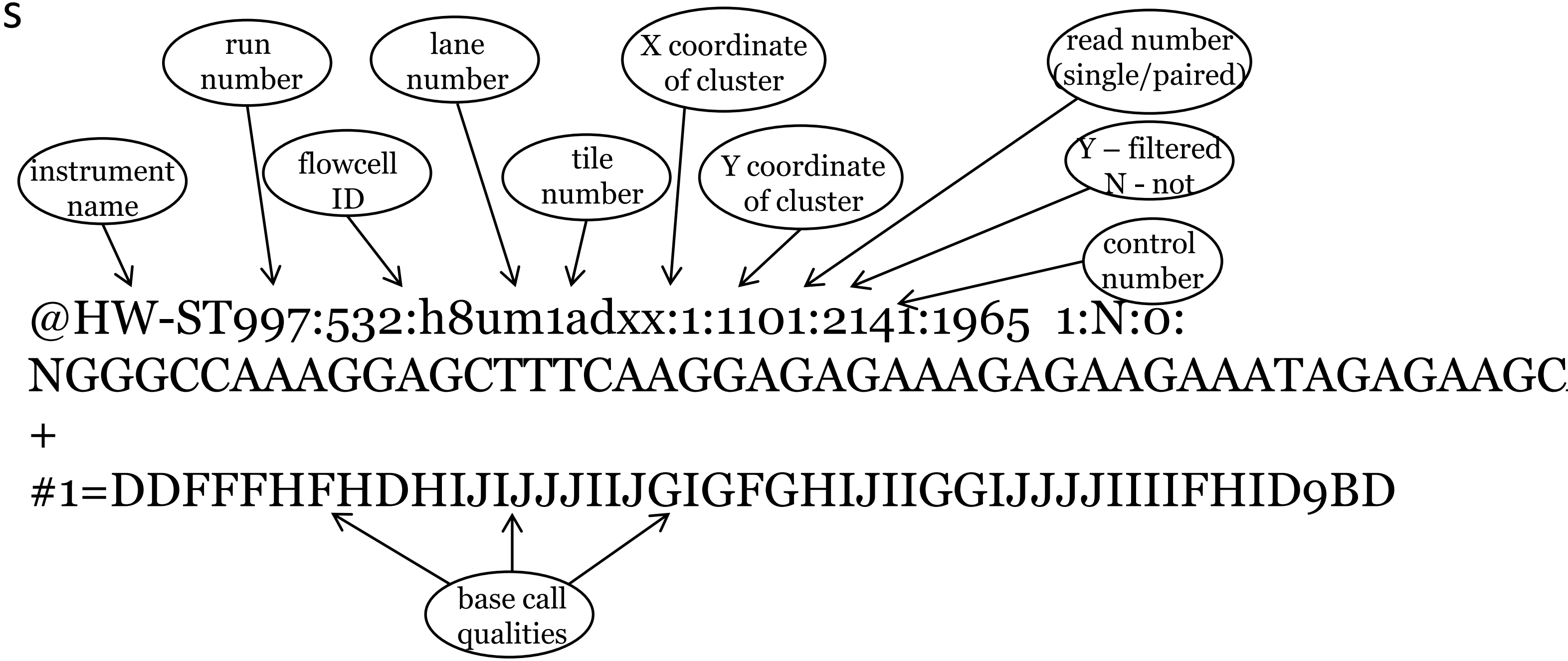
- In terminal:

  -
    ```
    cd ~/shared/RNAseq/GSE120762/downsampled
    ```

  -
    ```
    zcat SRR7956038_1.fastq.gz | head
    ```

Raw sequence file formats: FASTQ

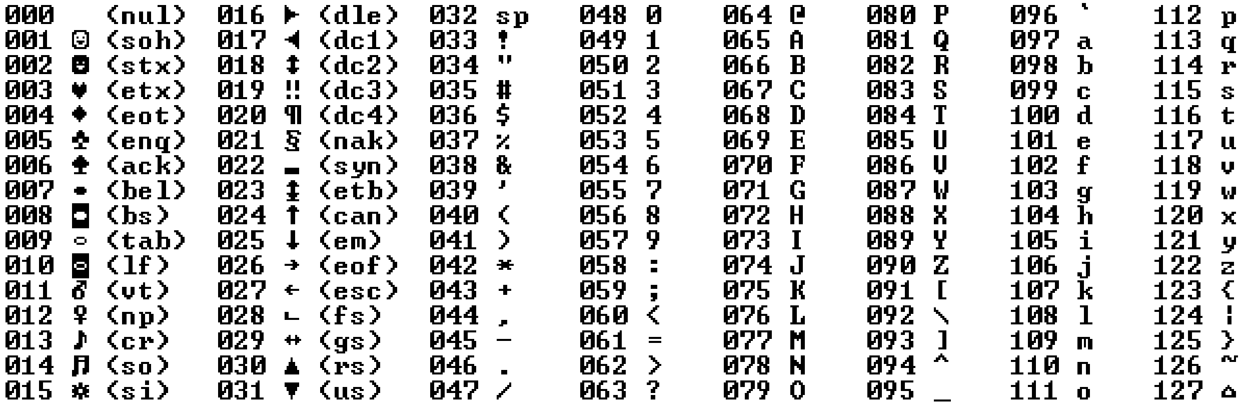
- FASTQ format has 4 lines per read: name, sequence, comment, base call qualities



```
@HW-ST997:532:h8um1adxx:1:1101:2141:1965  1:N:0:
NGGGCCAAAGGAGCTTTCAAGGAGAGAAAGAGAAGAAATAGAGAAGC
+
#1=DDFFFHFHDHIJIJJJIIJGIGFGHIJIIGGIJJJJIIIIFHID9BD
```

Base call qualities

- Phred qualities: **Q = -10logP**, where P is the probability of error
- Q (Phred score) scale: Q = 10, 90% acc; 30, 99.9% acc
- Sanger encoding: Phred qualities 0 to 93, then add 33 (33-126) and convert to ASCII characters (! or # is the lowest, ~ is highest)



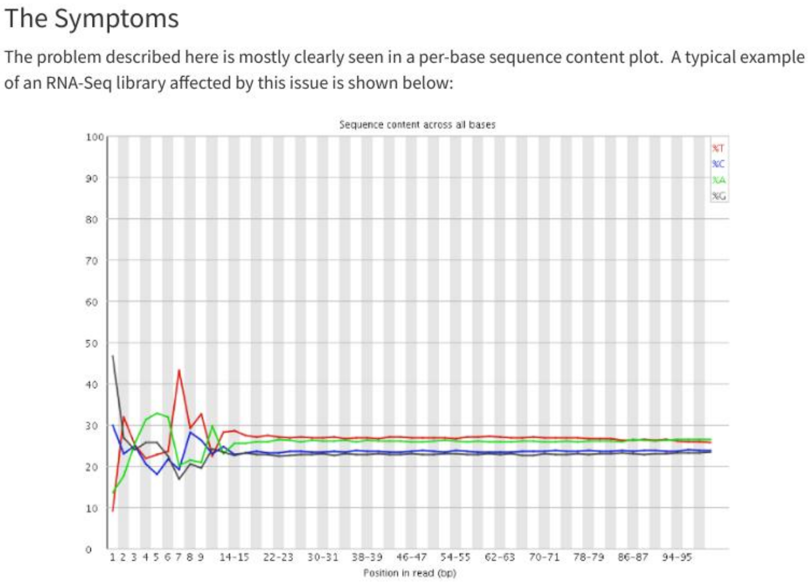
000		<nul>	016	►	<dle>	032	sp	048	0	064	@	080	P	096	`	112	p
001	☺	<soh>	017	◄	<dc1>	033	!	049	1	065	A	081	Q	097	a	113	q
002	☻	<stx>	018	↕	<dc2>	034	"	050	2	066	B	082	R	098	b	114	r
003	♥	<etx>	019	‼	<dc3>	035	#	051	3	067	C	083	S	099	c	115	s
004	♦	<eot>	020	¶	<dc4>	036	$	052	4	068	D	084	T	100	d	116	t
005	♣	<enq>	021	§	<nak>	037	%	053	5	069	E	085	U	101	e	117	u
006	♠	<ack>	022	▬	<syn>	038	&	054	6	070	F	086	V	102	f	118	v
007	•	<bel>	023	↨	<etb>	039	'	055	7	071	G	087	W	103	g	119	w
008	◘	<bs>	024	↑	<can>	040	(	056	8	072	H	088	X	104	h	120	x
009	○	<tab>	025	↓	<em>	041	)	057	9	073	I	089	Y	105	i	121	y
010	◙	<lf>	026	→	<eof>	042	*	058	:	074	J	090	Z	106	j	122	z
011	♂	<vt>	027	←	<esc>	043	+	059	;	075	K	091	[	107	k	123	{
012	♀	<np>	028	∟	<fs>	044	,	060	<	076	L	092	\	108	l	124	¦
013	♪	<cr>	029	↔	<gs>	045	-	061	=	077	M	093	]	109	m	125	}
014	♫	<so>	030	▲	<rs>	046	.	062	>	078	N	094	^	110	n	126	~
015	☼	<si>	031	▼	<us>	047	/	063	?	079	O	095	_	111	o	127	⌂

First steps

- ✓ **Go back** to the project gzdirectory open new terminal (Alt-Shif-R) or run
  - `cd ~/rnaseq`
- ✓ Open do.sh in file editor
- ✓ Run “Step 1” block: select the lines in the editor and press “Ctrl-Enter”
- ✓ “fastqs” directory should appear

Quality control: FastQC

- Run “Step 2”
- Wait for it to finish and open “fastqc/SRR7956038/SRR7956038_1_fastqc.html” file
- Designed for DNA-seq, so “bad” is not always bad
- QCFail: https://sequencing.qcfail.com/



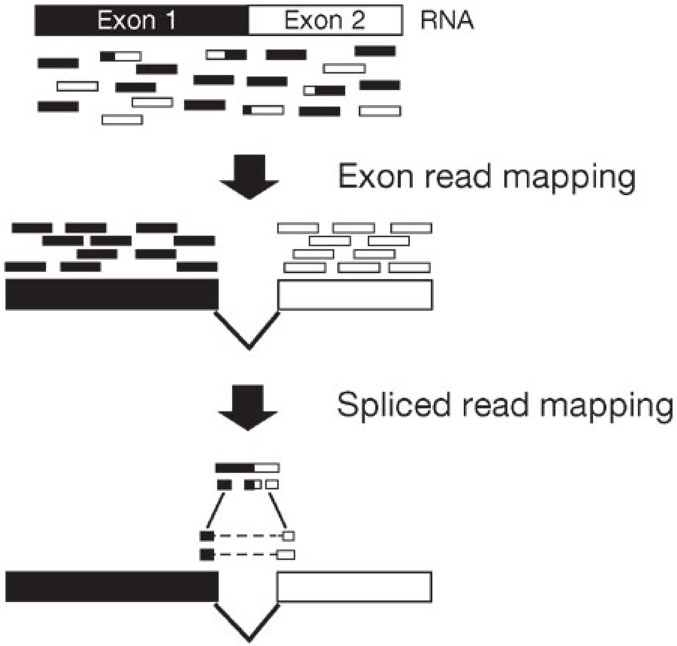
The Symptoms

The problem described here is mostly clearly seen in a per-base sequence content plot. A typical example of an RNA-Seq library affected by this issue is shown below:

https://sequencing.qcfail.com/articles/positional-sequence-bias-in-random-primed-libraries/

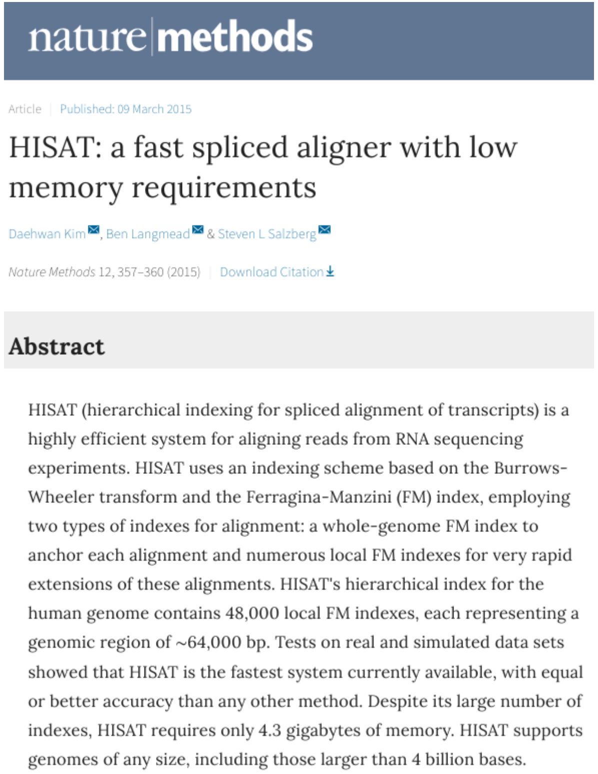
Alignment + counting pipeline

- ✓ Alignment
  - HISAT2
  - STAR
  - bowtie/bowtie2
- ✓ Counting
  - featureCounts
  - htseq
  - mmquant



Hisat2

- ✓ https://ccb.jhu.edu/software/hisat2/index.shtml
- ✓ Run “Step 3”



nature methods

Article | Published: 09 March 2015

HISAT: a fast spliced aligner with low memory requirements

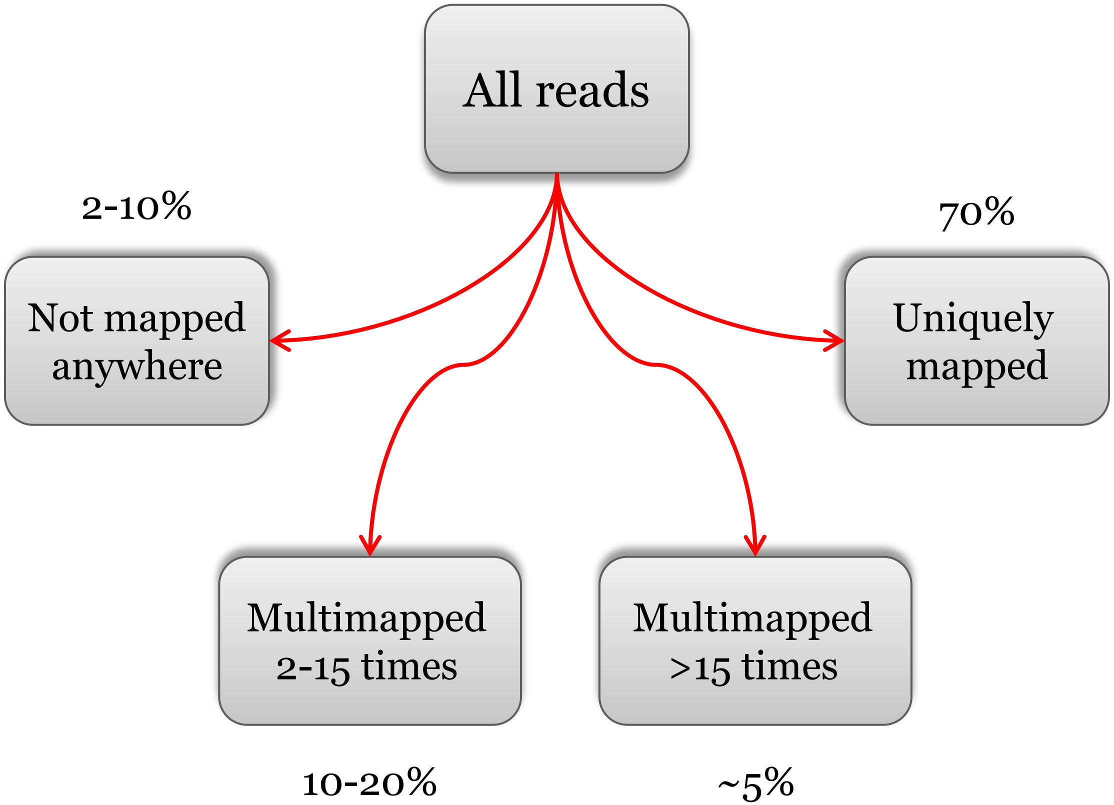
Daehwan Kim ✉, Ben Langmead ✉ & Steven L Salzberg ✉

Nature Methods 12, 357–360 (2015) | Download Citation ⤓

Abstract

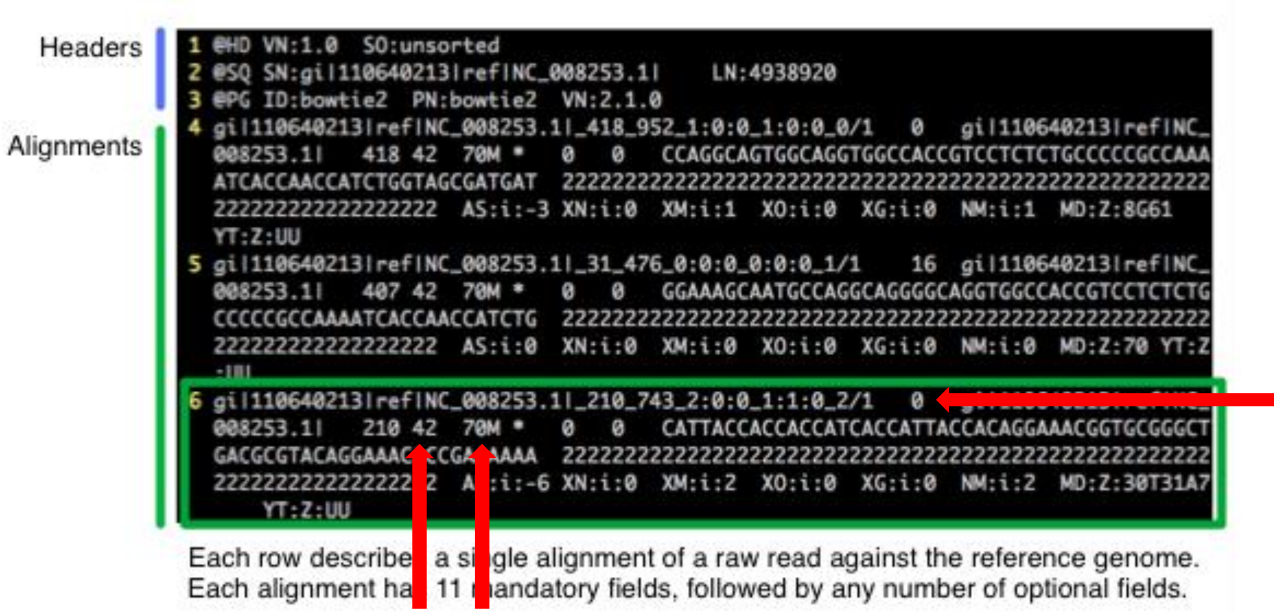
HISAT (hierarchical indexing for spliced alignment of transcripts) is a highly efficient system for aligning reads from RNA sequencing experiments. HISAT uses an indexing scheme based on the Burrows-Wheeler transform and the Ferragina-Manzini (FM) index, employing two types of indexes for alignment: a whole-genome FM index to anchor each alignment and numerous local FM indexes for very rapid extensions of these alignments. HISAT's hierarchical index for the human genome contains 48,000 local FM indexes, each representing a genomic region of ~64,000 bp. Tests on real and simulated data sets showed that HISAT is the fastest system currently available, with equal or better accuracy than any other method. Despite its large number of indexes, HISAT requires only 4.3 gigabytes of memory. HISAT supports genomes of any size, including those larger than 4 billion bases.

Expectations for genomic RNA-seq alignment



View the alginment

- Run “Step 3.5”



```
@HD VN:1.0  SO:unsorted
@SQ SN:gi|110640213|ref|NC_008253.1|    LN:4938920
@PG ID:bowtie2  PN:bowtie2  VN:2.1.0
gi|110640213|ref|NC_008253.1|_418_952_1:0:0_1:0:0_0/1    0    gi|110640213|ref|NC_
008253.1|    418 42   70M *    0    0    CCAGGCAGTGGCAGGTGGCCACCGTCCTCTCTGCCCCCGCCAAA
ATCACCAACCATCTGGTAGCGATGAT  22222222222222222222222222222222222222222222222222
22222222222222222222  AS:i:-3 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:8G61
YT:Z:UU
gi|110640213|ref|NC_008253.1|_31_476_0:0:0_0:0:0_1/1    16   gi|110640213|ref|NC_
008253.1|    407 42   70M *    0    0    GGAAAGCAATGCCAGGCAGGGGCAGGTGGCCACCGTCCTCTCTG
CCCCCGCCAAAATCACCAACCATCTG  22222222222222222222222222222222222222222222222222
22222222222222222222  AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:70 YT:Z
:UU
gi|110640213|ref|NC_008253.1|_210_743_2:0:0_1:1:0_2/1    0    gi|110640213|ref|NC_
008253.1|    210 42   70M *    0    0    CATTACCACCACCATCACCATTACCACAGGAAACGGTGCGGGCT
GACGCGTACAGGAAACACCGAAAAAA  22222222222222222222222222222222222222222222222222
22222222222222222222  AS:i:-6 XN:i:0  XM:i:2  XO:i:0  XG:i:0  NM:i:2  MD:Z:30T31A7
    YT:Z:UU
```

Each row describes a single alignment of a raw read against the reference genome. Each alignment has 11 mandatory fields, followed by any number of optional fields.

bitwise SAM flag

mapping quality

CIGAR string

Working with the alignemnt files: samtools

- view
- sort
- index
- …

- Run “Step 4”

Exercise

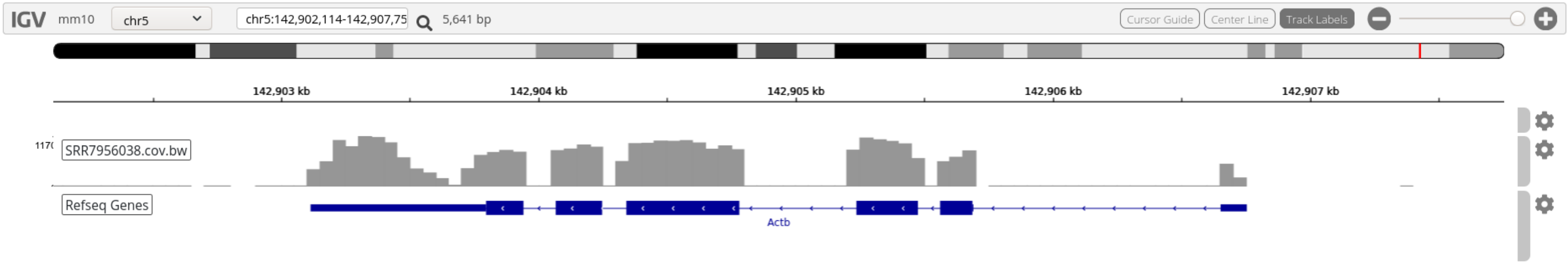
- What is the alignment rate if mapped to human genome?

Read coverage

- ✓ Run “Step 5”
- ✓ Formats
  - wig
  - **bigwig**
  - tdf
- ✓ Download SRR7956938.cov.bw file:
  - click on checkbox on the right
  - select More/Export…

View read coverage

- Go to IGV genome browser at https://igv.org/app/
- Select GRCm38/mm10 genome
- Load a track from SRR7956938.cov.bw file
- Go to Actb gene (type "Actb" and press "Enter")

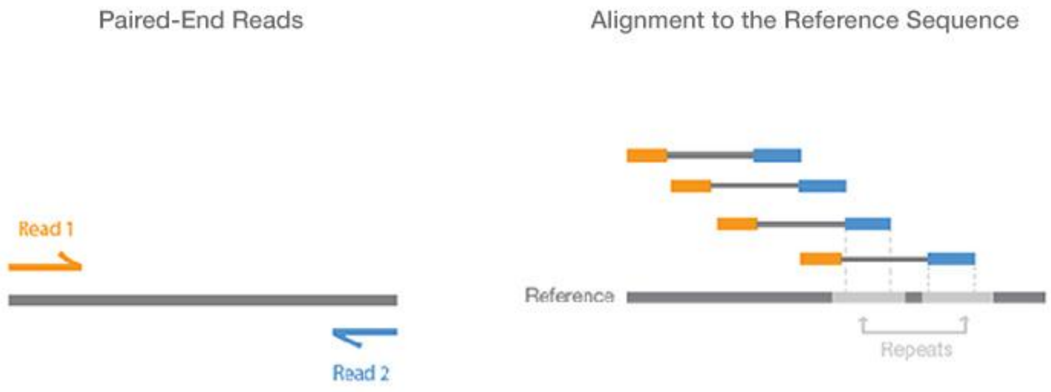


mm10 chr5 chr5:142,902,114-142,907,75 5,641 bp

RNAseq quality control

- ✓ Run “Step 6”
- ✓ RSeQC (http://rseqc.sourceforge.net)
  - infer_experiment.py
  - read_distribution.py
  - geneBody_coverage.py
- ✓ Picard
  - CollectRnaSeqMetrics
- ✓ Useful to check ribosomal RNA content as well

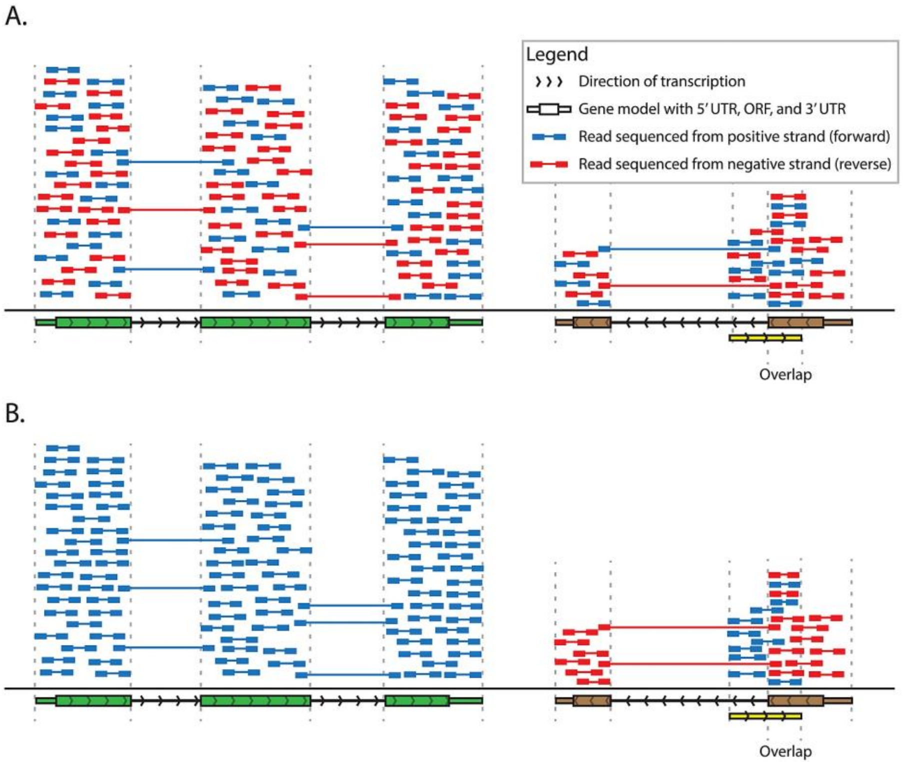
Library strategies: single-end vs paired-end



Paired-end sequencing enables both ends of the DNA fragment to be sequenced. Because the distance between each paired read is known, alignment algorithms can use this information to map the reads over repetitive regions more precisely. This results in much better alignment of the reads, especially across difficult-to-sequence, repetitive regions of the genome.

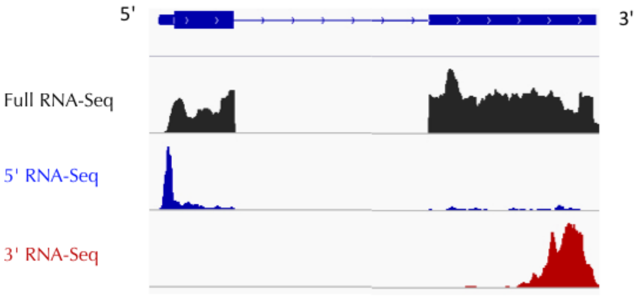
https://www.illumina.com/science/technology/next-generation-sequencing/plan-experiments/paired-end-vs-single-read.html

Library strategies: stranded vs unstranded

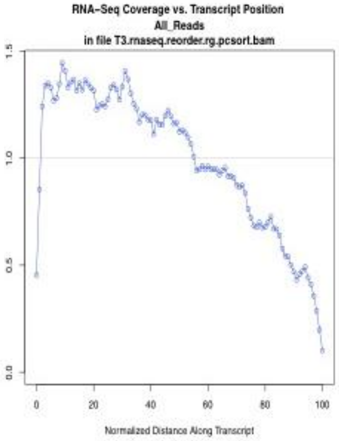

A.
Legend
››› Direction of transcription
Gene model with 5′ UTR, ORF, and 3′ UTR
Read sequenced from positive strand (forward)
Read sequenced from negative strand (reverse)
Overlap
B.
Overlap

https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004393

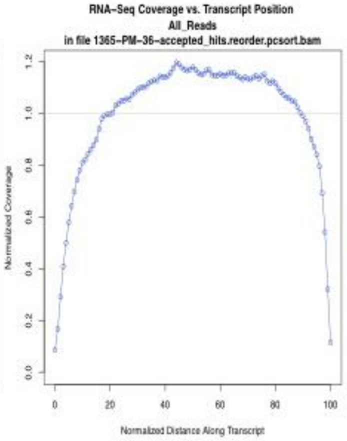
Library strategies: 3’- or 5’- specific, full-length



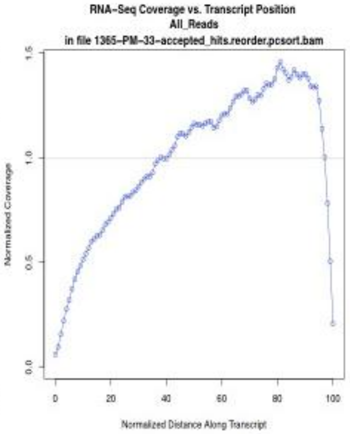
5’ bias
(strand oriented protocol)



no bias
(low coverage at ends
of transcript)



3’ bias
(poly-A selection)

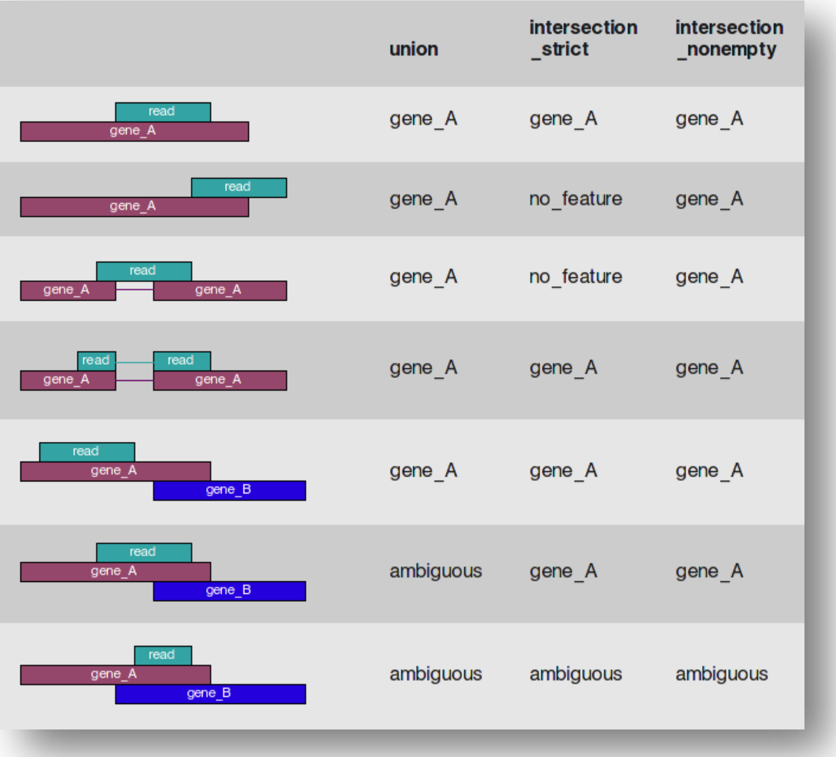


Zuojian Tang NYU Langone MEDICAL CENTER

https://bitesizebio.com/13559/ngs-quality-control-in-rna-sequencing-some-free-tools/

FeatureCounts

- Run “Step 7”
- For stranded experiment, we can distinguish between two different genes if they are on the opposite strands
- For non-stranded experiment, we can’t
- htseq-count discards reads with 2 or more features (ambiguous)
- ~50% assignment rate is normal



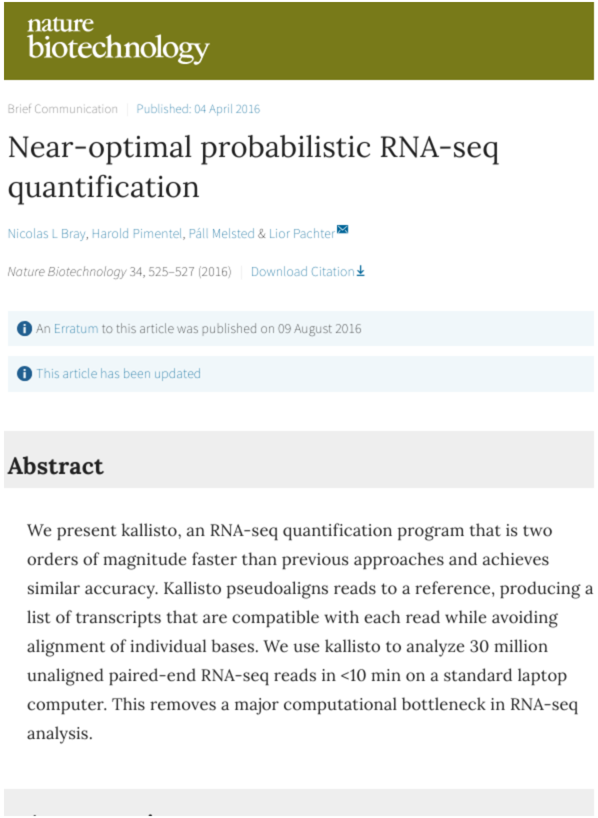
	union	intersection_strict	intersection_nonempty
read gene_A	gene_A	gene_A	gene_A
read gene_A	gene_A	no_feature	gene_A
read gene_A gene_A	gene_A	no_feature	gene_A
read read gene_A gene_A	gene_A	gene_A	gene_A
read gene_A gene_B	gene_A	gene_A	gene_A
read gene_A gene_B	ambiguous	gene_A	gene_A
read gene_A gene_B	ambiguous	ambiguous	ambiguous

Library depth

- >5M assigned reads are required for a typical analysis, thus there should be >10M raw reads
- Usually it’s better to increase the number of biological replicates instead of library depth

Kallisto

- ✓ Run “Step 8”
- ✓ Pseudo-alignment
- ✓ No sam/bam output
- ✓ Transcript level quantification
- ✓ Expectation-maximization for counting multimappers/ambigous reads



nature biotechnology

Brief Communication | Published: 04 April 2016

Near-optimal probabilistic RNA-seq quantification

Nicolas L Bray, Harold Pimentel, Páll Melsted & Lior Pachter

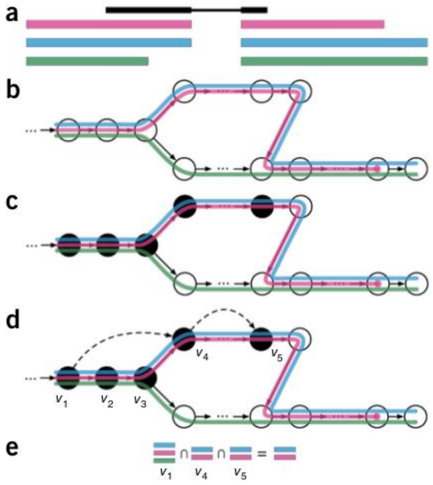
Nature Biotechnology 34, 525–527 (2016) | Download Citation

An Erratum to this article was published on 09 August 2016

This article has been updated

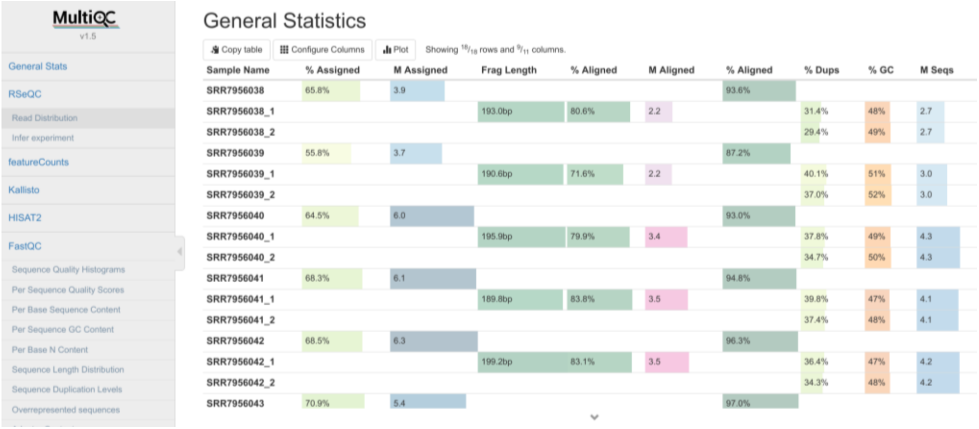
Abstract

We present kallisto, an RNA-seq quantification program that is two orders of magnitude faster than previous approaches and achieves similar accuracy. Kallisto pseudoaligns reads to a reference, producing a list of transcripts that are compatible with each read while avoiding alignment of individual bases. We use kallisto to analyze 30 million unaligned paired-end RNA-seq reads in <10 min on a standard laptop computer. This removes a major computational bottleneck in RNA-seq analysis.



MultiQC

- Generate a single report for many tools:
  - fastqc
  - hisat2
  - rseqc
  - kallisto
  - …
- Run “Step 9”
- Open the report: multiqc_report.html



General Statistics

Showing $^{18}/_{18}$ rows and $^{9}/_{11}$ columns.

Sample Name	% Assigned	M Assigned	Frag Length	% Aligned	M Aligned	% Aligned	% Dups	% GC	M Seqs
SRR7956038	65.8%	3.9				93.6%			
SRR7956038_1			193.0bp	80.6%	2.2		31.4%	48%	2.7
SRR7956038_2							29.4%	49%	2.7
SRR7956039	55.8%	3.7				87.2%			
SRR7956039_1			190.6bp	71.6%	2.2		40.1%	51%	3.0
SRR7956039_2							37.0%	52%	3.0
SRR7956040	64.5%	6.0				93.0%			
SRR7956040_1			195.9bp	79.9%	3.4		37.8%	49%	4.3
SRR7956040_2							34.7%	50%	4.3
SRR7956041	68.3%	6.1				94.8%			
SRR7956041_1			189.8bp	83.8%	3.5		39.8%	47%	4.1
SRR7956041_2							37.4%	48%	4.1
SRR7956042	68.5%	6.3				96.3%			
SRR7956042_1			199.2bp	83.1%	3.5		36.4%	47%	4.2
SRR7956042_2							34.3%	48%	4.2
SRR7956043	70.9%	5.4				97.0%			

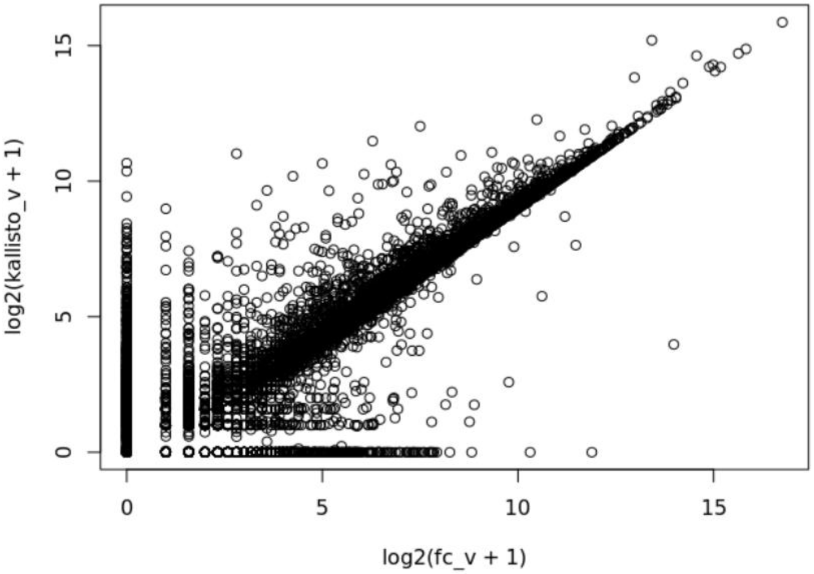
CPM/FPKM/RPKM/TPM

- ✓ CPM = Counts per Million
- ✓ FPKM = Fragments per Kilobase of gene per Million
- ✓ RPKM = Reads per Kilobase of gene per Million
  - different from FPKM if library is paired-end
- ✓ TPM = Transcripts per Kilobase Million
  - first normalize to gene (transcript) length, then to library depth

https://www.rna-seqblog.com/rpkm-fpkm-and-tpm-clearly-explained/

Importing kallisto

- ✓ Open do.R
- ✓ Run everything



Exercise

- ✓ Compare kallisto vs featureCounts
  - what genes are highly different between kallisto and featureCounts?

Repeating for all the samples

- ✓ Open do_all.sh
  - **don’t run!**
- ✓ Go to ~/shared/RNAseq/GSE120762/results_all/ in the file explorer
- ✓ Open multiqc_report.html

Summary

- ✓ Know your reference genomes and annotations
- ✓ Know your library prep
- ✓ Went from raw data to gene expression tables
  - Alignment + quantification pipeline
  - Alignment-free analysis with kallisto
- ✓ QC for every step