Bioinformatics

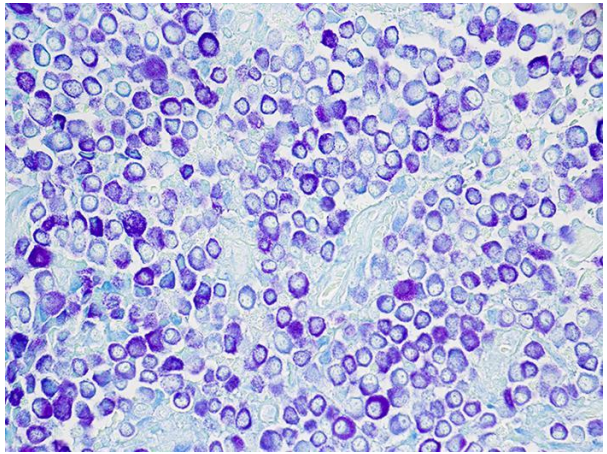
Brian Arnold

Overview for next 3 lectures

- Intro to bioinformatics
  - Focus on whole-genome sequencing
  - Discussion of other data types
- Run whole-genome sequencing pipeline on Princeton's HPC
  - Simple, made from scratch
- Converting the above pipeline into a 'snakemake' workflow

what is bioinformatics?

- analysis of biological data
  - **DNA**
  - **RNA**
  - proteins
  - metabolites



- in mixture (bulk), in single cells, or across space

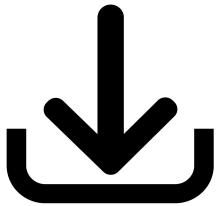
- in biology departments, bioinformatics essentially involves understanding, downloading, and running other people's software to analyze your data
- *sometimes* custom code is required (i.e. manipulating output files, plotting results), but majority of data analysis involves existing software

what is bioinformatics ‘pipeline’ or ‘workflow’?

- a series of programs run in order, so that the output of the first program flows into the second program, etc…

Bioinformatics involves generally useful skills

- searching what tools are best/popular/appropriate for your data
  - look at methods section of other papers that do similar things
  - google/chatGPT
  - sometimes multiple options, but programs that are highly used and cited are usually easy to use and give useful output
- reading the documentation carefully to see if they do what you need
  - what options or ‘arguments’ do the programs accept?
- downloading and installing all these programs
  - mamba or conda is *absolutely essential*
- getting them to work on your computer or on Princeton’s HPC
  - conda environments should take care of this most of the time!
  - google/chatGPT the error message



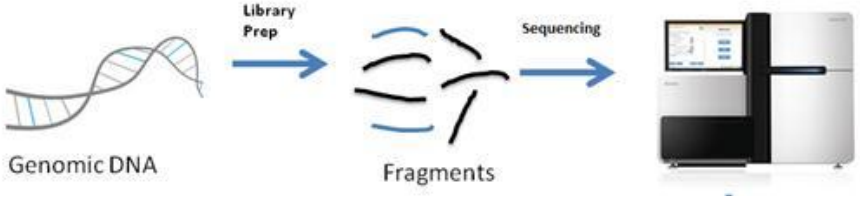
- bioinformatics just involves doing this on biological data, typically something generated by one of the various sequencing technologies

An example: whole-genome sequencing (WGS)

that we'll run on the cluster next time, step by step

and after that, fully automate in a 'snakemake' pipeline

mutations



file type

RAW FASTQ → CLEANED FASTQ → BAM → BAM deduplicated → VCF

Clean raw data

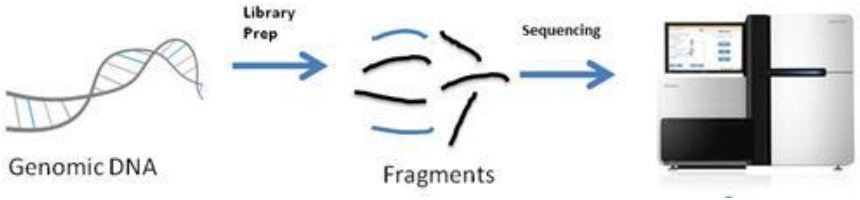
Align sequencing reads

Remove PCR duplicated

Find mutations

process in pipeline

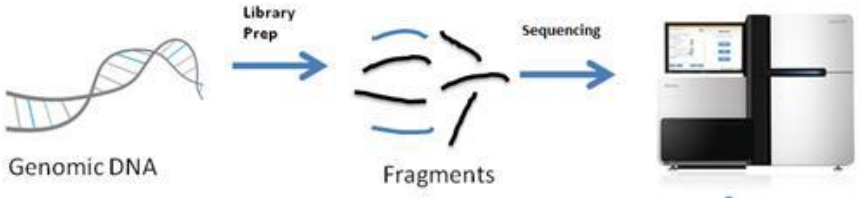
WGS summary: analyzing mutations

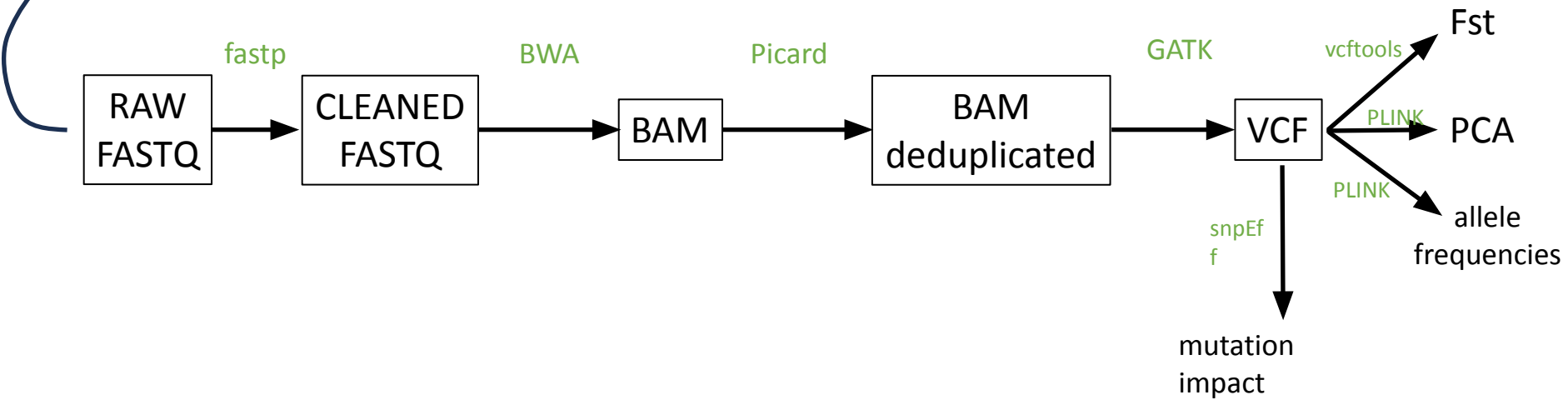


RAW FASTQ → CLEANED FASTQ → BAM → BAM deduplicated → VCF

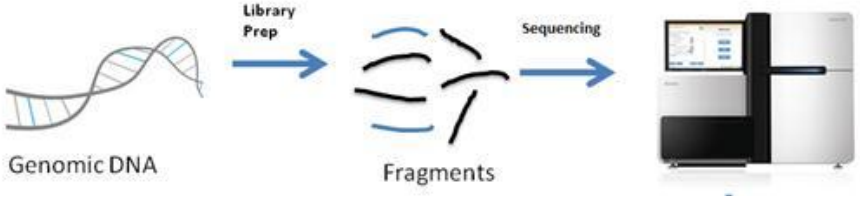
VCF → Fst
VCF → PCA
VCF → allele frequencies
VCF → mutation impact

WGS summary: programs to get there





WGS summary: programs to get there

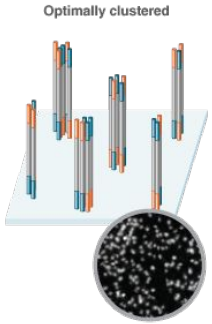


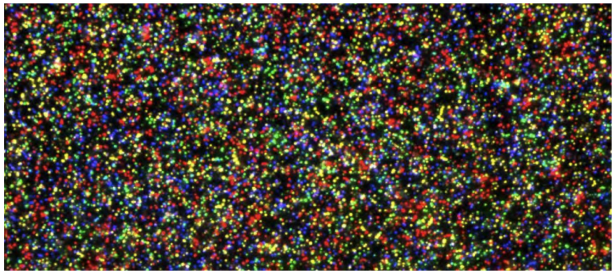
fastp BWA Picard GATK

RAW FASTQ → CLEANED FASTQ → BAM → BAM deduplicated → VCF

Let's focus on this part first

FASTQ file





Colors are converted to nucleotides in a FASTQ file!

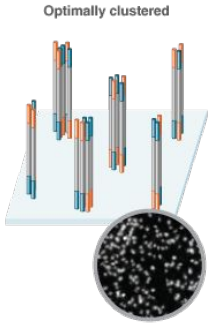
Here's one entry of a FASTQ file (typically there are millions):

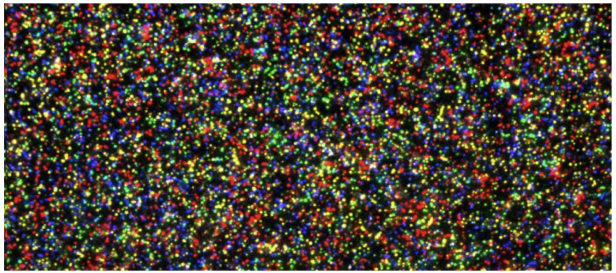
```
@A00351:121:HHW2DSXX:1:1101:1416:1000 1:N:0:GCTCTGTA+NATCCAAG
TTCATTTTTTTTTTTTTTTGATGCTCAGAAGAGTCTTTTTTCCTGATATGCAGTGCCTTTGGGATGGTAGCACTATACCAGGTGTCAGCCTGCATTCTGGTTTGAACTTTAGAATGTTTCAGTTTTCTTTTTTAATGGGGATTGCGATGGT
+
FFFFFFFFFFFFFFFFFF::,:FF,F,FF,FFF,FFFFFF,F:F::FF,FF,F,FF:FF,:FF,,FFF,FF,F,,F:FF,FF:F:F,FF:F:F,F,::FFFF,:F,FFFF,FF,,FFF,F:FFFF,:FFF,FFFFFFFF:FF:FFFFFF
```

4 lines per sequencing read

1. identifier
2. **DNA sequence**
3. some separator
4. base quality scores (coded using ASCII characters to represent numerical scores)

CLEANED FASTQ: throw away bad data





Each nucleotide in FASTQ has a quality score

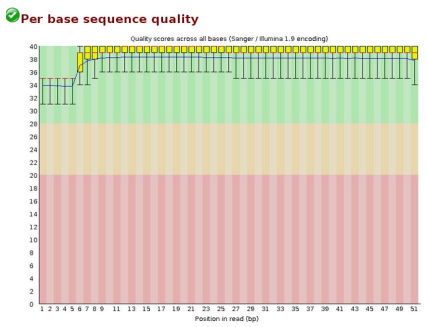
trim nucleotides from reads

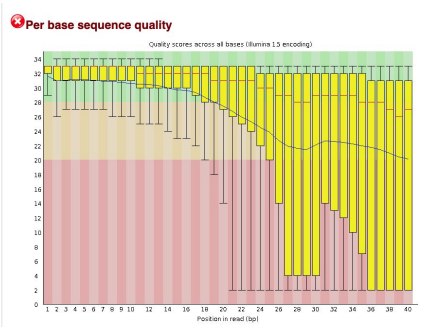
- Illumina adapter sequences
- poly-A tails (if RNAseq data)
- low quality nucleotides at **end** of reads

discard entire reads that have many low quality nucleotides

Good programs:

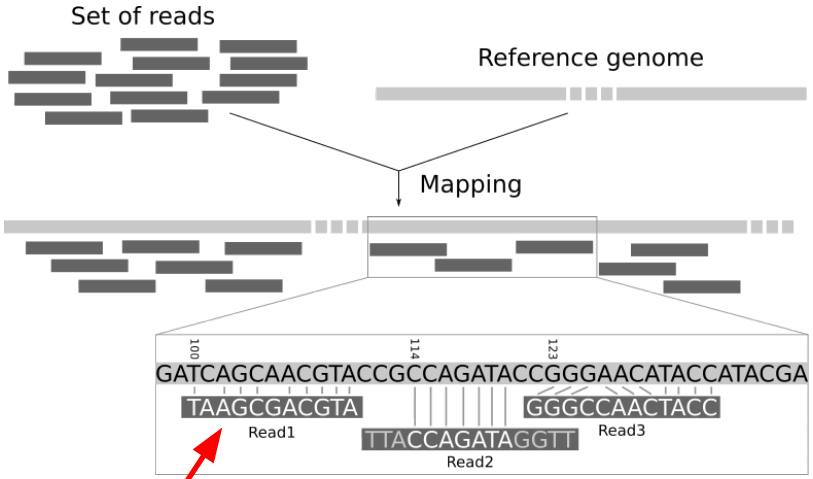
- fastp
- fastqc





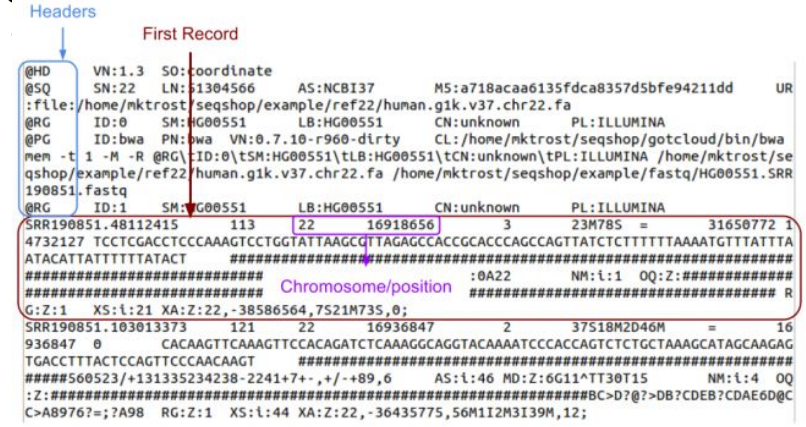
BAM file: reference genome alignment

- align or map reads to a known reference genome
- best if reads and reference genome are **from the same species**
  - Otherwise, hard to find similarities b/t reads and reference!
  - some programs specialize in aligning to diverged reference



Reads get aligned best on **similarity** to positions in reference genome

Good programs:



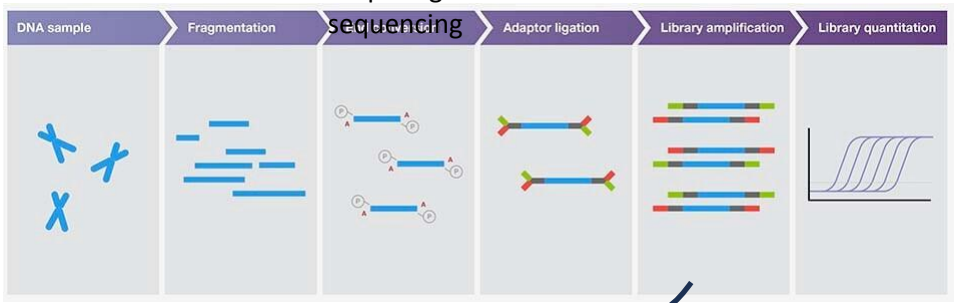

```
@HD     VN:1.3  SO:coordinate
@SQ     SN:22   LN:51304566     AS:NCBI37       M5:a718acaa6135fdca8357d5bfe94211dd     UR
:file:/home/mktrost/seqshop/example/ref22/human.g1k.v37.chr22.fa
@RG     ID:0    SM:HG00551      LB:HG00551      CN:unknown      PL:ILLUMINA
@PG     ID:bwa  PN:bwa  VN:0.7.10-r960-dirty    CL:/home/mktrost/seqshop/gotcloud/bin/bwa
mem -t 1 -M -R @RG\tID:0\tSM:HG00551\tLB:HG00551\tCN:unknown\tPL:ILLUMINA /home/mktrost/se
qshop/example/ref22/human.g1k.v37.chr22.fa /home/mktrost/seqshop/example/fastq/HG00551.SRR
190851.fastq
@RG     ID:1    SM:HG00551      LB:HG00551      CN:unknown      PL:ILLUMINA
SRR190851.48112415      113     22      16918656        3       23M78S  =       31650772 1
4732127 TCCTCGACCTCCCAAAGTCCTGGTATTAAGCGTTAGAGCCACCGCACCCAGCCAGTTATCTCTTTTTTAAAATGTTTATTTA
ATACATTATTTTTTATACT     ##############################################################################
############################                    :0A22       NM:i:1  OQ:Z:##############
############################ ################################### R
G:Z:1   XS:i:21 XA:Z:22,-38586564,7S21M73S,0;
SRR190851.103013373     121     22      16936847        2       37S18M2D46M     =       16
936847  0       CACAAGTTCAAAGTTCCACAGATCTCAAAGGCAGGTACAAAATCCCACCAGTCTCTGCTAAAGCATAGCAAGAG
TGACCTTTACTCCAGTTCCCAACAAGT     ############################################################
#####560523/+131335234238-2241+7+-,+/-+89,6      AS:i:46 MD:Z:6G11^TT30T15       NM:i:4  OQ
:Z:#############################################################BC>D?@?>DB?CDEB?CDAE6D@C
C>A8976?=;?A98  RG:Z:1  XS:i:44 XA:Z:22,-36435775,56M1I2M3I39M,12;
```

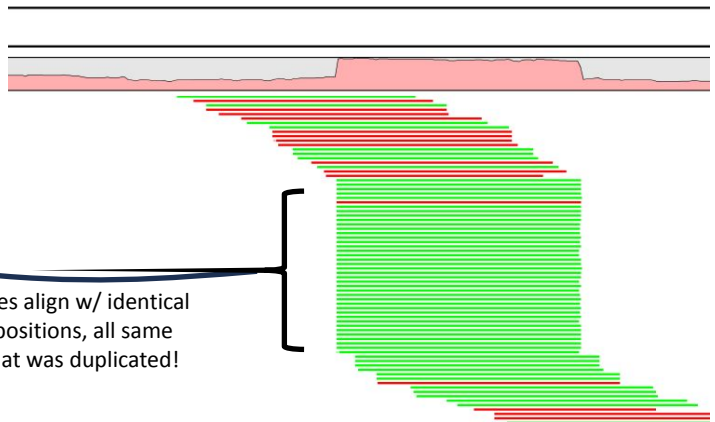
BAM deduplicated: throw away more bad data

mark PCR duplicates

- most analyses assume short DNA fragments (~500bp) have been randomly sampled across the genome
- in many DNA preparation protocols, there is a PCR step to
  - enrich DNA fragments with adapters
  - increase amount of DNA
  - introduce extra barcodes to the ends of DNA to track samples
- PCR duplicates should be identified so downstream programs are aware of them!

Preparing DNA for sequencing





PCR duplicates align w/ identical start/stop positions, all same molecule that was duplicated!

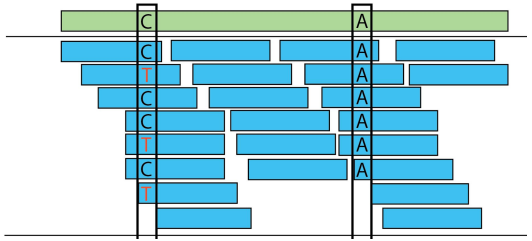
Good programs:

- picard
- sambamba

VCF file: variant calling

- use reference-aligned reads to detect single nucleotide polymorphisms (SNPs)
  - nucleotides that differ from the reference genome sequence

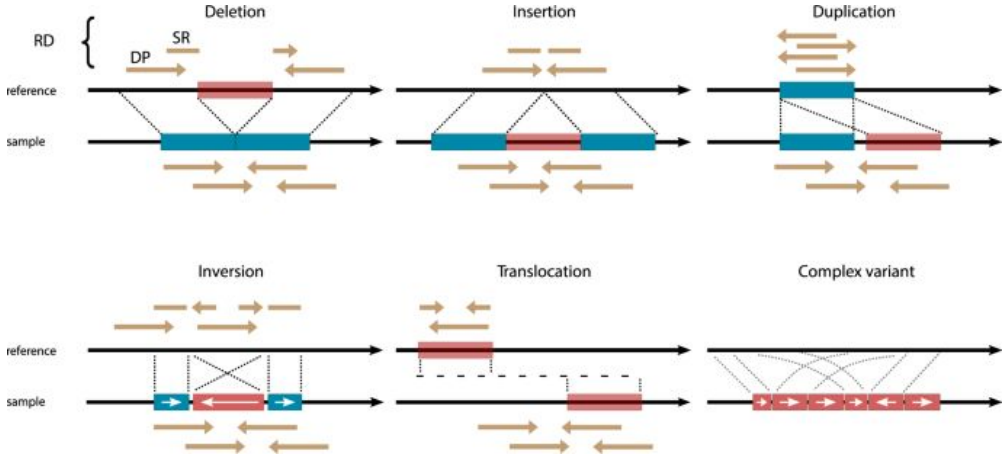
Good programs:
- GATK4 (gold standard but painful to use)
- freebayes



looks like a C/T heterozygote

variant calling (indels, copy number, structural)

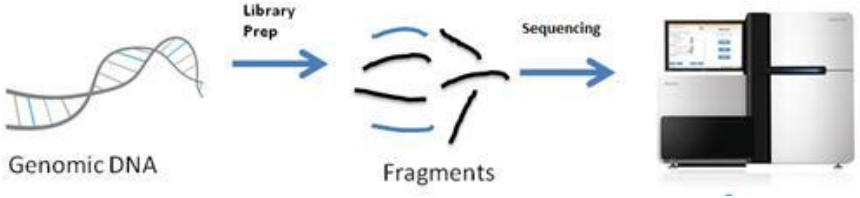
You can also detect larger scale differences



Good programs:

- GATK4 for short events
- DELLY, LUMPY, MANTA
- Sniffles (long read)

WGS summary: programs to get there

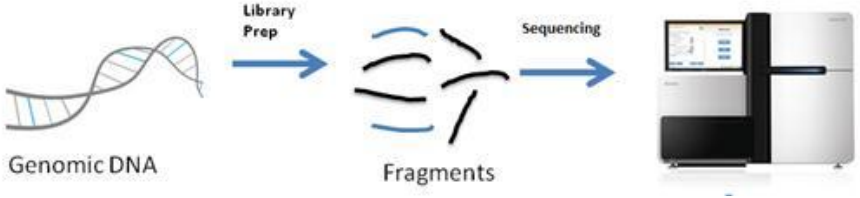


fastp BWA Picard GATK

RAW FASTQ → CLEANED FASTQ → BAM → BAM deduplicated → VCF

What we just covered

WGS summary: programs to get there



Now let's focus on this part

RAW FASTQ → CLEANED FASTQ → BAM → BAM deduplicated → VCF

- VCF → Fst (vcftools)
- VCF → PCA (PLINK)
- VCF → allele frequencies (PLINK)
- VCF → mutation impact (snpEff)

VCF files: brief description

```
##fileformat=VCFv4.3
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,md5=f126cdf8a6e0c7f379d618ff66beb2da,species="Homo sapiens",taxonomy=x>
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS     ID        REF  ALT    QUAL FILTER  INFO                              FORMAT      NA00001        NA00002        NA00003
20     14370   rs6054257 G    A      29   PASS    NS=3;DP=14;AF=0.5;DB;H2           GT:GQ:DP:HQ 0|0:48:1:51,51 1|0:48:8:51,51  1/1:43:5:.,.
20     17330   .         T    A      3    q10     NS=3;DP=11;AF=0.017               GT:GQ:DP:HQ 0|0:49:3:58,50 0|1:3:5:65,3    0/0:41:3
20     1110696 rs6040355 A    G,T    67   PASS    NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:21:6:23,27 2|1:2:0:18,2    2/2:35:4
20     1230237 .         T    .      47   PASS    NS=3;DP=13;AA=T                   GT:GQ:DP:HQ 0|0:54:7:56,60 0|0:48:4:51,51  0/0:61:2
20     1234567 microsat1 GTC  G,GTCT 50   PASS    NS=3;DP=9;AA=G                    GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
```

- understanding VCF files is important! many analyses you do start here

- Header
  - Starts with '##': contains many useful definitions of abbreviations throughout file
  - Starts with '#CHROM': column names for each variant
    - E.g. chromosome, position, REF and ALT allele
    - 'FORMAT' column explains how information displayed for samples columns, which follow FORMAT column

VCF files: brief description

```
##fileformat=VCFv4.3
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,md5=f126cdf8a6e0c7f379d618ff66beb2da,species="Homo sapiens",taxonomy=x>
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS     ID        REF  ALT    QUAL FILTER  INFO                              FORMAT      NA00001        NA00002        NA00003
20     14370   rs6054257 G    A      29   PASS    NS=3;DP=14;AF=0.5;DB;H2           GT:GQ:DP:HQ 0|0:48:1:51,51 1|0:48:8:51,51  1/1:43:5:.,.
20     17330   .         T    A      3    q10     NS=3;DP=11;AF=0.017               GT:GQ:DP:HQ 0|0:49:3:58,50 0|1:3:5:65,3    0/0:41:3
20     1110696 rs6040355 A    G,T    67   PASS    NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:21:6:23,27 2|1:2:0:18,2    2/2:35:4
20     1230237 .         T    .      47   PASS    NS=3;DP=13;AA=T                   GT:GQ:DP:HQ 0|0:54:7:56,60 0|0:48:4:51,51  0/0:61:2
20     1234567 microsat1 GTC  G,GTCT 50   PASS    NS=3;DP=9;AA=G                    GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
```

- FORMAT
  - Info separated by colons ‘:’
  - GT stands for genotype:
    - 0/0 are homozogotes for the REF allele
    - 1/1 are homozogotes for the ALT allele
    - 0/1 have both alleles! Heterozygotes!
  - DP stands for depth (or sequencing depth, how many reads covered that position)
    - Anything above 8 is considered gold, helps ensures homozygotes not actually heterozygotes
    - E.g. If only 1 read covers a position, it’s impossible to know if it’s a heterozygote!

VCF files: brief description

```
##fileformat=VCFv4.3
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,md5=f126cdf8a6e0c7f379d618ff66beb2da,species="Homo sapiens",taxonomy=x>
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS      ID         REF   ALT    QUAL  FILTER   INFO                              FORMAT       NA00001         NA00002          NA00003
20     14370    rs6054257  G     A      29    PASS     NS=3;DP=14;AF=0.5;DB;H2           GT:GQ:DP:HQ  0|0:48:1:51,51  1|0:48:8:51,51   1/1:43:5:.,.
20     17330    .          T     A      3     q10      NS=3;DP=11;AF=0.017               GT:GQ:DP:HQ  0|0:49:3:58,50  0|1:3:5:65,3     0/0:41:3
20     1110696  rs6040355  A     G,T    67    PASS     NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ  1|2:21:6:23,27  2|1:2:0:18,2     2/2:35:4
20     1230237  .          T     .      47    PASS     NS=3;DP=13;AA=T                   GT:GQ:DP:HQ  0|0:54:7:56,60  0|0:48:4:51,51   0/0:61:2
20     1234567  microsat1  GTC   G,GTCT 50    PASS     NS=3;DP=9;AA=G                    GT:GQ:DP     0/1:35:4        0/2:17:2         1/1:40:3
```

- Quality control (which can be done in vcftools)
  - Discard sites in which many individuals have too low of depth
  - Discard entire samples that have many missing

VCF files analysis: intro

- I have a VCF file of tusked and tuskless elephants
- How many positions do I have mutation information for? Count number of lines in file, but exclude those beginning

```
(base) [bjarnold@argo-comp2 VCF]$ grep -v '#' elephants.vcf | wc
-l
```

- if this number is 0 you have a problem
  :)

- When analyzing VCF files, I frequently look at the last line of the header which contains all the individual/sample names
  1. are there as many samples as you expect? sample names follow the FORMAT column
  2. are the sample names what you expect?

```
(base) [bjarnold@argo-comp2 VCF]$ grep CHROM elephants.vcf
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  0045B   2981B   2982B   2983B   2984B   2985B
2986A   G13     G15     G16     G17A    G18A    G19A    G20A    G21A    G22A    T2B
```

VCF file analysis: Fst

Let's use **vcftools** to calculate Fst in sliding windows along the first chromosome

- FST is calculated between two *groups* of individuals, where there are many samples within a group
- If groups are different species, Fst will be high
- If groups are within species, Fst will generally be lower

We need to tell vcftools which samples in our VCF file correspond to group 1 and group 2!

tusked.txt

```
2981
B
2982
B
2985
B
G18A
```

tuskless.txt

```
0045
B
2983
B
2984
B
2986
A
G17A
G19A
G22A
```

VCF file analysis: Fst

command line usage of

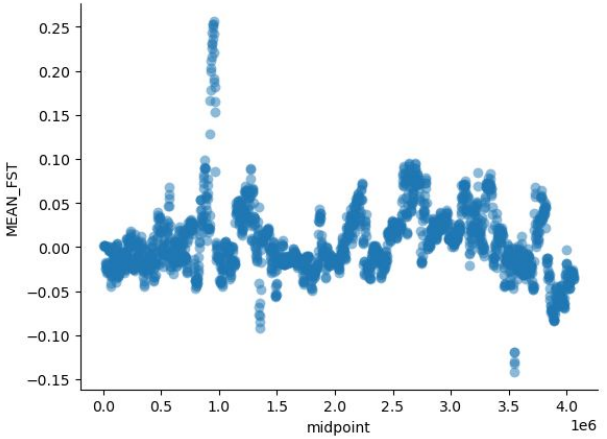
```
VCF=/path/to/my/VCF/elephants.vcf

vcftools --vcf ${VCF} \
--weir-fst-pop tuskless.txt \
--weir-fst-pop tusked.txt \
--fst-window-size 10000 \
--fst-window-step 2000 \
--out fst_output
```

example

CHROM	BIN_START	BIN_END	N_VARIANTS	WEIGHTED_FST	MEAN_FST
scaffold_0	1	10000	27	0.00637893	0.00169378
scaffold_0	2001	12000	28	0.0111226	0.00206346
scaffold_0	4001	14000	25	0.000697218	0.00127949
scaffold_0	6001	16000	30	0.00413623	-4.60565e-05
scaffold_0	8001	18000	33	0.01503	0.00171518
scaffold_0	10001	20000	30	0.0172676	0.00150262
scaffold_0	12001	22000	26	0.00319997	-0.0177068
scaffold_0	14001	24000	30	0.000517151	-0.019309

Plot 6th column with python or R



VCF file analysis: PCA

command line usage of

```
VCF=/path/to/my/VCF/elephants.vcf

plink --vcf ${VCF} \
--pca 2 \
--allow-extra-chr \
--out pca_output
```

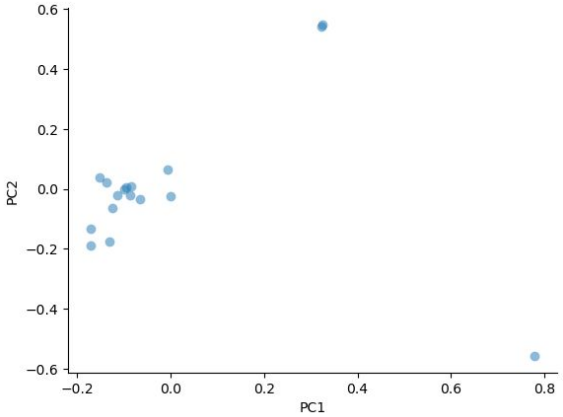
output file:

```
0045B 0045B 0.321888 0.54021
2981B 2981B -0.000314435
-0.025959
2982B 2982B -0.137171 0.0209842
2983B 2983B -0.0647252 -0.0345524
2984B 2984B -0.171759 -0.187896
2985B 2985B -0.169974 -0.133993
2986A 2986A 0.324437 0.548219
G13 G13 -0.129983 -0.176786
G15 G15 -0.0873685 -0.0219399
```

Plot last 2 columns in python or R

results meaningless b/c

- only looking at small region in genome
- samples come from same population



VCF file analysis: allele frequencies

command line usage of

```
VCF=/path/to/my/VCF/elephants.vcf

plink --vcf ${VCF} \
--freq \
--allow-extra-chr \
--out allele_freqs
```

output file:

CHR	SNP	A1	A2	MAF	NCHROBS
scaffold_0	.	T	A	0.2647	34
scaffold_0	.	G	A	0.2647	34
scaffold_0	.	T	C	0.08824	34
scaffold_0	.	G	T	0.08824	34
scaffold_0	.	T	C	0.2647	34
scaffold_0	.	C	G	0.2647	34
scaffold_0	.	A	T	0.08824	34
scaffold_0	.	GC	G	0.2647	

There were 17 *diploid* samples in this VCF, or 34 chromosomes
for each sample, count number of T alleles

- heterozygotes contribute 1 T allele
- homozygotes contribute 2 T alleles

9/34 = 0.2647
9 chromosomes has A1 allele (T) and, 25 chromosomes had A2 allele (A)

VCF file analysis: annotation

```
7 117227832 . G T . . AC 14 AN 22
ANN
T|stop_gained|HIGH|CFTR|ENSG00000001626|transcript|ENST00000003084|protein_coding|12
/27|c.1624G>T|p.Gly542*|1756/6128|1624/4443|542/1480||
ANN
T|stop_gained|HIGH|CFTR|ENSG00000001626|transcript|ENST00000454343|protein_coding|11
/26|c.1441G>T|p.Gly481*|1573/5949|1441/4260|481/1419||
LOF (CFTR|ENSG00000001626|11|0.27)
NMD (CFTR|ENSG00000001626|11|0.27)
```

Good programs

- SNPEff
  - has *many* species annotations you can download

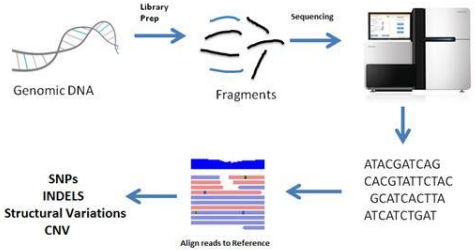
Exporting VCF files to simpler table

```
mamba activate gatk4
gatk VariantsToTable \
-V elephants.vcf \
-F CHROM -F POS -F TYPE -GF GT \
-O output.table
```

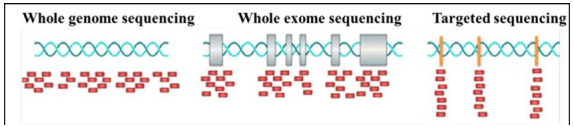
CHROM	POS	TYPE	0045B.GT	2981B.GT	2982B.GT	2983B.GT	2984B.GT
scaffold_0	145	SNP	A/A	A/A	A/T	A/T	A/T
scaffold_0	396	SNP	A/A	A/A	A/G	A/G	A/G
scaffold_0	412	SNP	C/C	C/C	C/C	C/T	C/C
scaffold_0	530	SNP	C/C	C/C	C/T	C/T	C/T
scaffold_0	538	SNP	G/G	G/G	G/C	G/C	G/C
scaffold_0	784	INDEL	G/G	G/G	G/GC	G/GC	G/GC
scaffold_0	1153	SNP	A/A	A/A	A/G	A/G	A/G
scaffold_0	1202	SNP	G/G	G/G	G/A	G/A	G/A

Other kinds of sequencing data

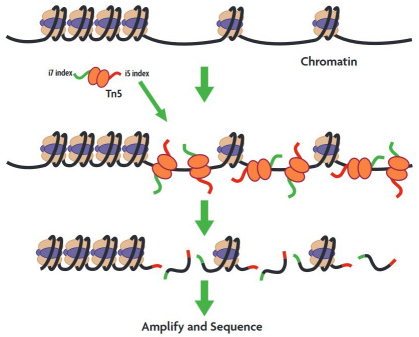
bioinformatics data: DNA



- Whole-genome sequencing (WGS)
  - find variants:
    - single-nucleotide polymorphisms (SNPs)
    - structural variants (e.g. DNA rearrangements)
    - copy number variants (e.g. gene duplications/deletions)
  - assemble new genome

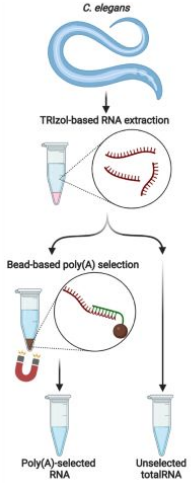


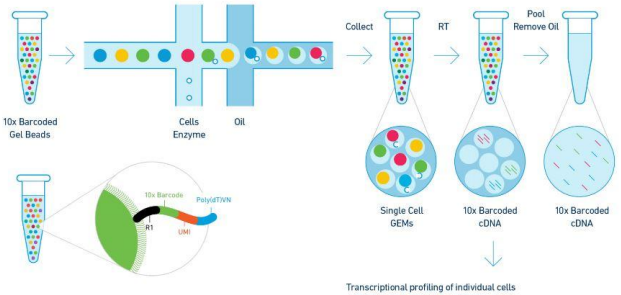
- Restriction-associated DNA (RADseq)
  - sequence DNA near restriction enzyme site
- ATACseq
  - sequence regions of the genome that are “open”
- CHIPseq
  - find regions where a protein binds to DNA

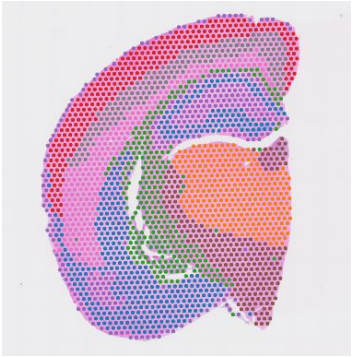


bioinformatics data

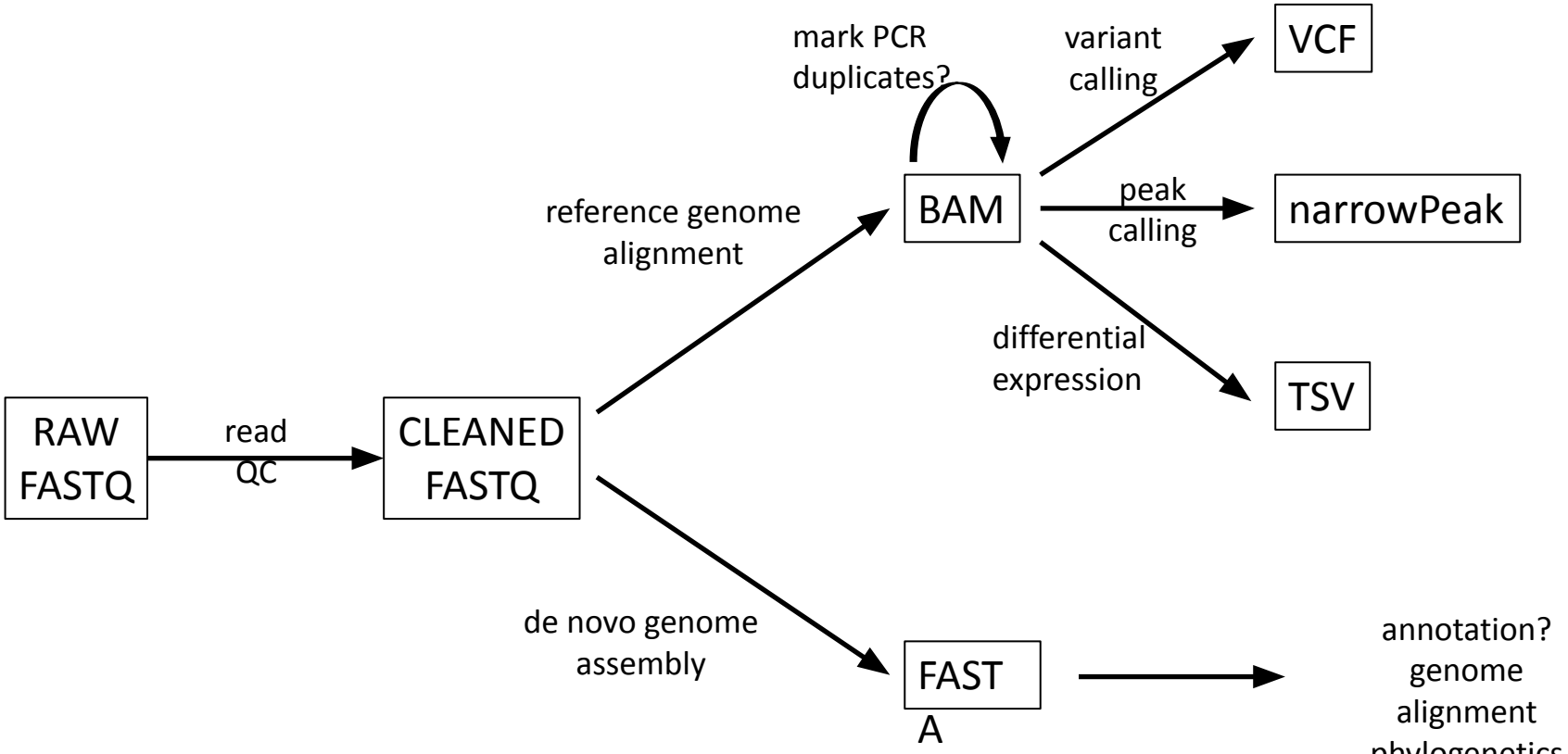
- RNA sequencing examples
  - RNAseq
    - actually DNA sequencing b/c converted to cDNA
    - differential expression, gene regulatory networks
  - single-cell RNAseq
    - characterize cell types
  - spatial transcriptomics
    - identify tissues, inter-tissue communication







bioinformatics tools roadmap



RNA seq workflow (brief)

RNAseq

trim reads to remove low quality bases, polyA sequences, illumina adapters etc.

- fastp
- trimmomatic

align reads to reference genome (splice aware) to get BAM file

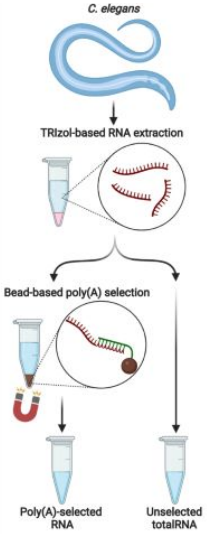
- STAR
- HISAT

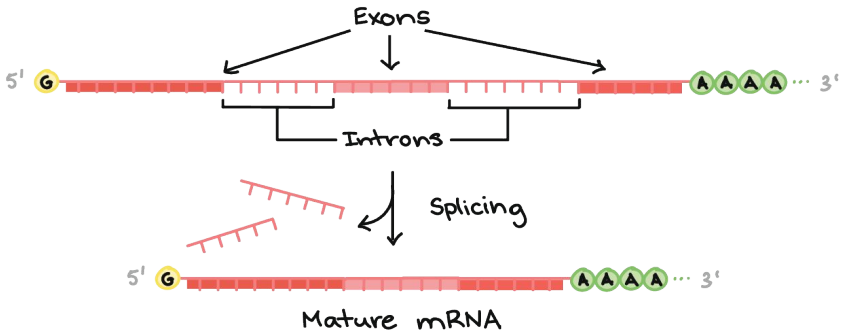
convert BAM to raw counts

- feature counts

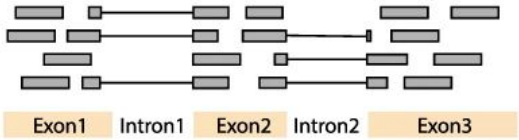
analyze count data

- deseq2
- edgeR

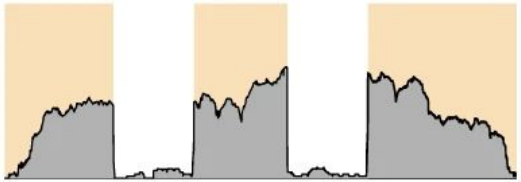




BAM file



Base-level expression coverage



RNAseq summary