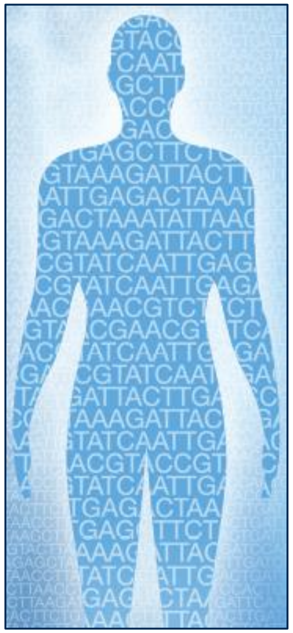
DNA Sequencing



Algoritmos Avançados de Bioinformática

Information transfer in the cell: nucleic acids

- Both DNA and RNA are polymers composed of four nucleic acid units, called nucleotides or bases.

- ❑ Adenyne (**A**) and Guanine (**G**), belong to one group (purynes).
- ❑ Cytosine (**C**) and Timine (**T**) and Uracil (**U**), belong to another group (pyrimidine).

- Timine only exists in DNA and Uracil is only found in RNA, the other three bases exist in both.

The **DNA** is composed of **two complementary strands** due to connections established between the bases in both strands.

- ❑ **Adenine and Timine** (A == T), connected by two hydrogen connections
- ❑ **Guanine and Cytosine** (G === C), connected by three hydrogen connections

- Chains are antiparallel because they are connected in opposite directions

Organization of genetic material

- **Genome**: an organism's genetic material (complete set of DNA)
    - a bacteria contains about 600,000 DNA base pairs
    - human and mouse genomes have some 3 billion.
    - human genome has 24 distinct chromosomes.
    - Each chromosome contains many genes.
- **Gene**: a discrete units of hereditary information located on the chromosomes and consisting of DNA and encode instructions on how to make proteins.
- **Genotype**: The genetic makeup of an organism
- **Phenotype**: the physical expressed traits of an organism

Organization of genetic material

Genome organization

Prokaryotic

- exists in the form of a circular chromosome located in the cytoplasm.

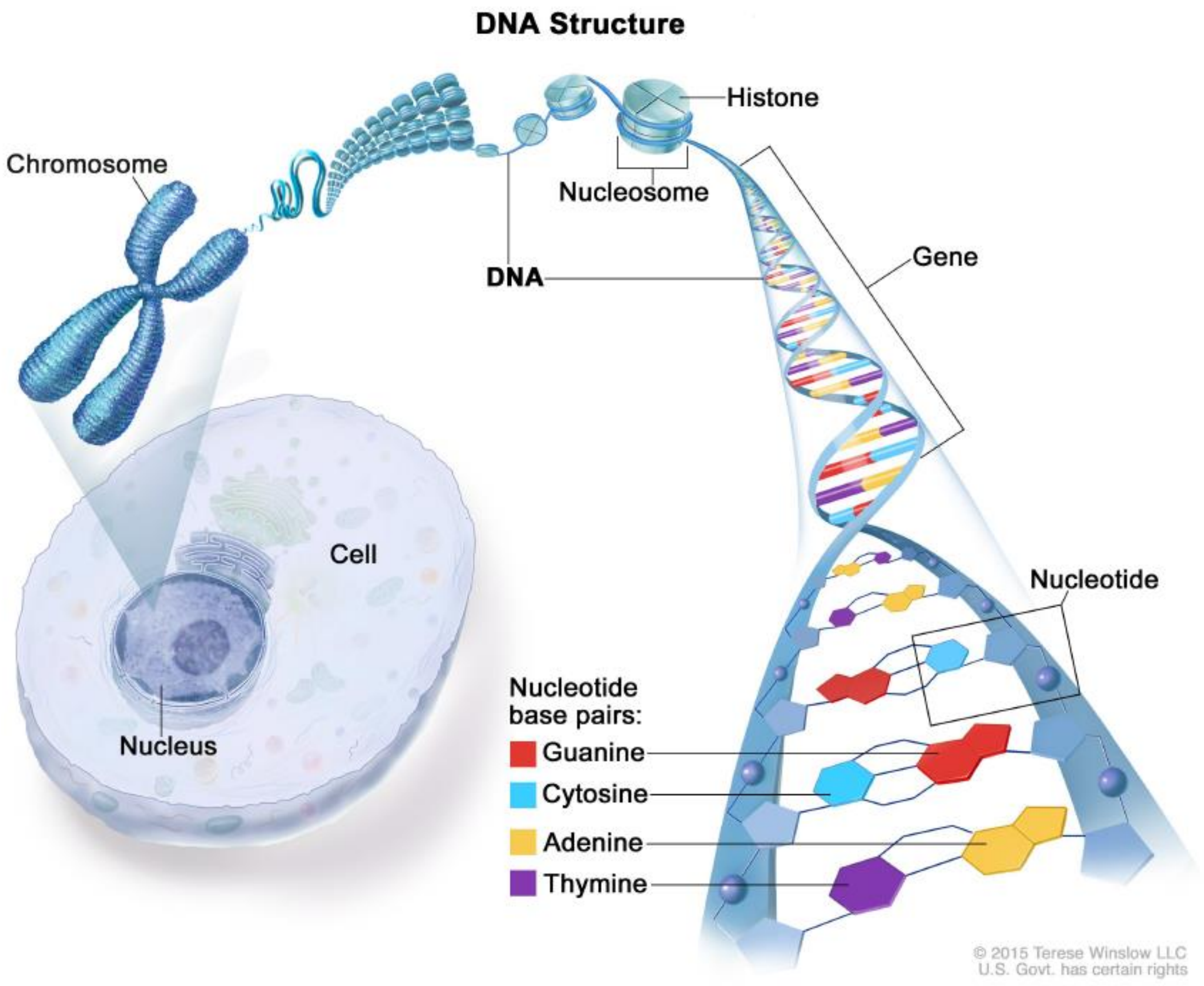
Eukaryotes

- found in the **nucleus** and
- tightly packaged into linear **chromosomes**.
- chromosomes consist of a DNA-protein complex called **chromatin** that is organized into subunits called **nucleosomes**.

This organization allows to:

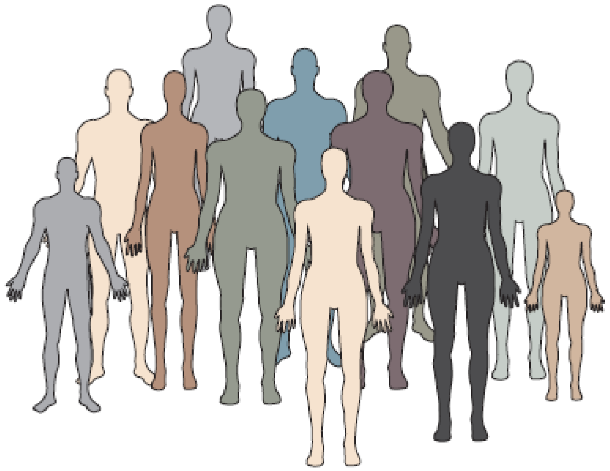
- fit a long DNA molecule in a small space
- provide regulatory structure for gene expression

Organization of genetic material



www.cancergenome.gov

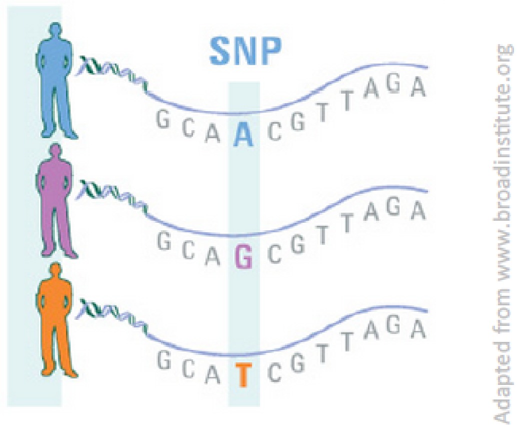
Human genome



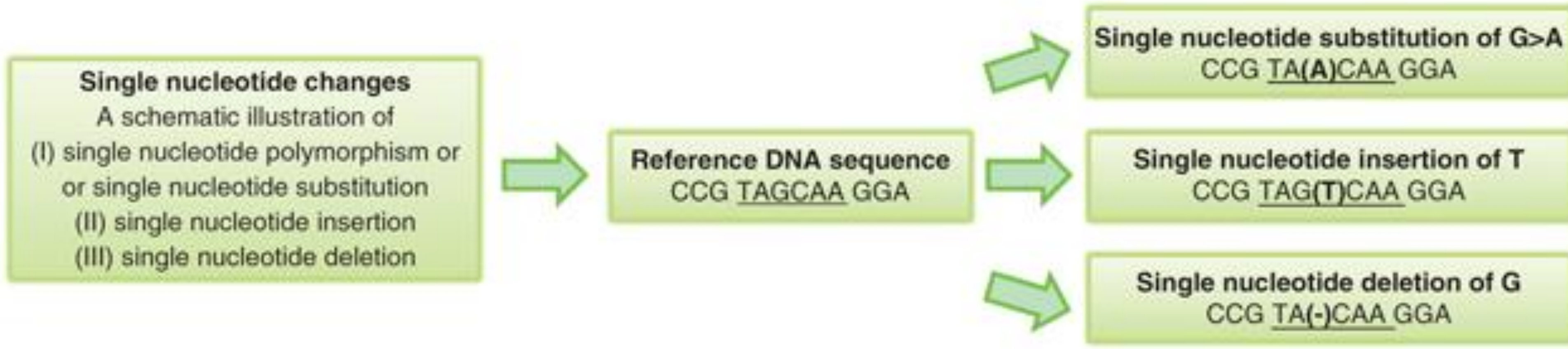
- Humans can be very different in aspect but we're all very similar at the DNA level.
- At the genetic level we are all very similar, with **more than 99% in common** with each other.
- This tiny fraction of genomic variation is very important and what make us unique.
- They determine the color of your eyes, hair and skin.
- They also influence your risk of disease and your response to drugs.

Single Nucleotide Polymorphism

- A single nucleotide polymorphism, or SNP.
- Variation at a single position in a DNA sequence among individuals.
- If a SNP occurs within a gene, then the gene is described as having more than one allele.
- Some SNPs (not all) may be associated with certain diseases.
- Scientists look for SNPs to evaluate an individual's genetic predisposition to develop a disease.

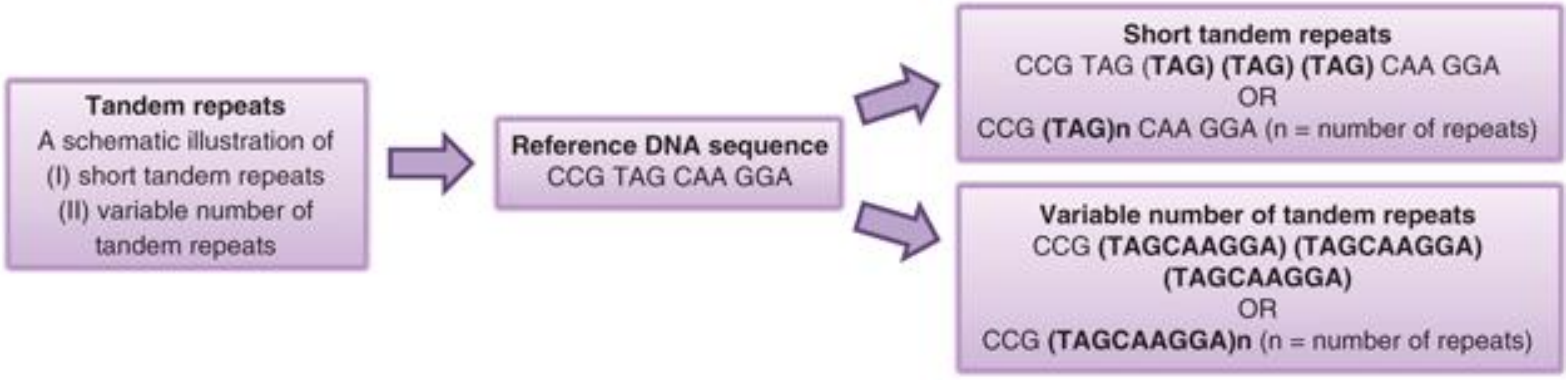


SNP Single Nucleotide Changes


Single nucleotide changes
A schematic illustration of
(I) single nucleotide polymorphism or
or single nucleotide substitution
(II) single nucleotide insertion
(III) single nucleotide deletion
Reference DNA sequence
CCG TAGCAA GGA
Single nucleotide substitution of G>A
CCG TA(A)CAA GGA
Single nucleotide insertion of T
CCG TAG(T)CAA GGA
Single nucleotide deletion of G
CCG TA(-)CAA GGA

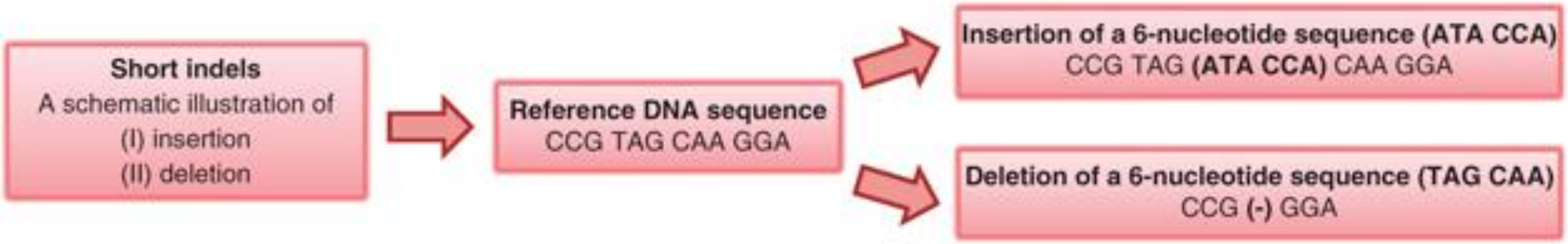
Adapted from C.Ku et al, " the discovery of human genetic variations and their use as disease markers: past, present and future" Journal of Human Genetics, 2017

Tandem Repeats


Tandem repeats
A schematic illustration of
(I) short tandem repeats
(II) variable number of
tandem repeats
Reference DNA sequence
CCG TAG CAA GGA
Short tandem repeats
CCG TAG **(TAG) (TAG) (TAG)** CAA GGA
OR
CCG **(TAG)n** CAA GGA (n = number of repeats)
Variable number of tandem repeats
CCG **(TAGCAAGGA) (TAGCAAGGA)**
(TAGCAAGGA)
OR
CCG **(TAGCAAGGA)n** (n = number of repeats)

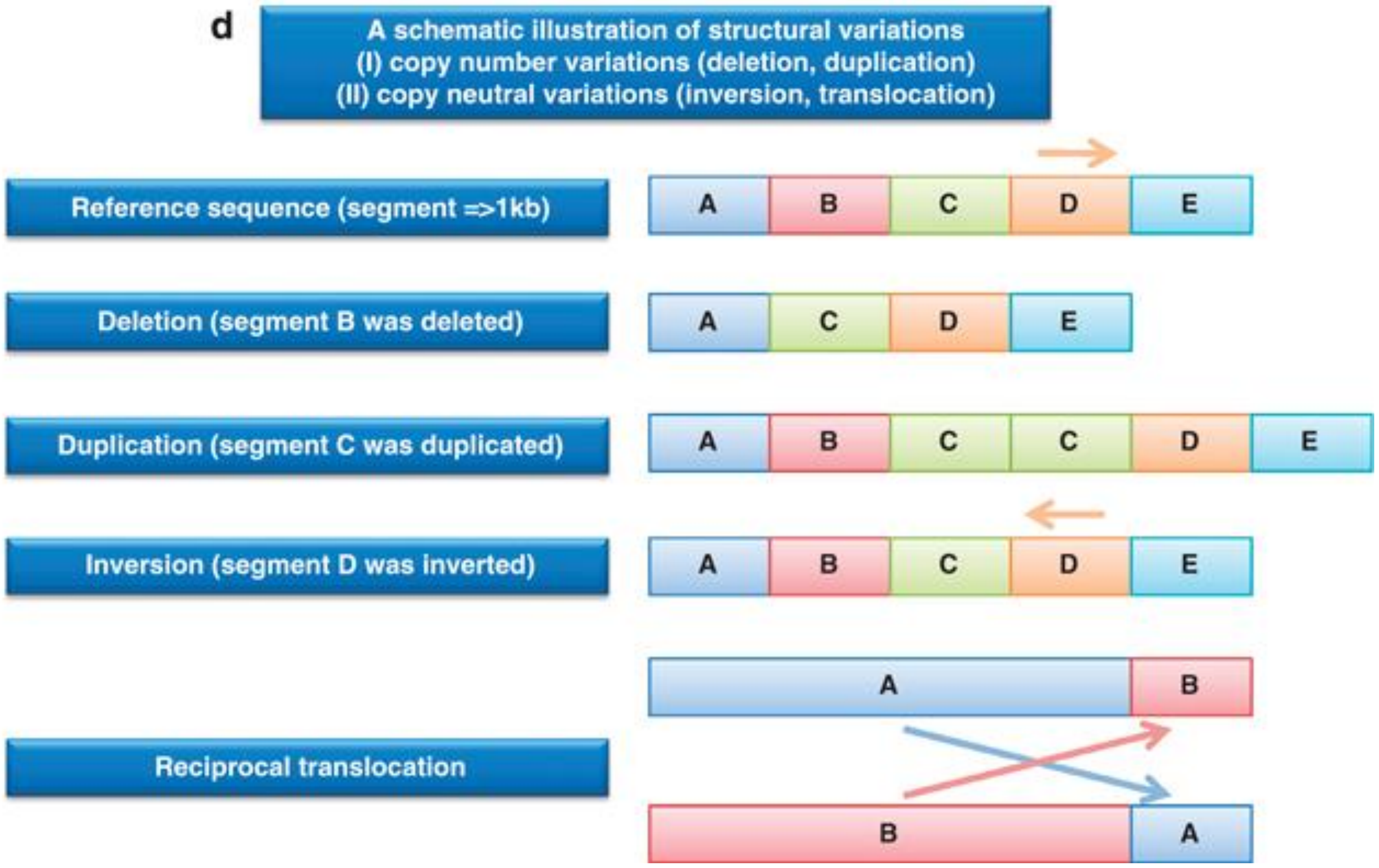
Adapted from C.Ku et al, " the discovery of human genetic variations and their use as disease markers: past, present and future" Journal of Human Genetics, 2017

Short indels


Short indels
A schematic illustration of
(I) insertion
(II) deletion
Reference DNA sequence
CCG TAG CAA GGA
Insertion of a 6-nucleotide sequence (ATA CCA)
CCG TAG **(ATA CCA)** CAA GGA
Deletion of a 6-nucleotide sequence (TAG CAA)
CCG **(-)** GGA

Adapted from C.Ku et al, " the discovery of human genetic variations and their use as disease markers: past, present and future" Journal of Human Genetics, 2017

Variations


d
A schematic illustration of structural variations
(I) copy number variations (deletion, duplication)
(II) copy neutral variations (inversion, translocation)
Reference sequence (segment =>1kb)
A B C D E
Deletion (segment B was deleted)
A C D E
Duplication (segment C was duplicated)
A B C C D E
Inversion (segment D was inverted)
A B C D E
Reciprocal translocation
A B
B A

Adapted from C.Ku et al, " the discovery of human genetic variations and their use as disease markers: past, present and future" Journal of Human Genetics, 2017

Human Genome Project: Frequently Asked Questions

Adapted from The Human Genome Project Completion: Frequently Asked Questions
https://www.genome.gov/11006943/human-genome-project-completion-frequently-asked-questions/

What is a genome?

- A genome is an organism's complete set of deoxyribonucleic acid (DNA).
- DNA molecules are made of two twisting, paired strands.
- Each strand is made of four chemical units, called nucleotide bases.
- The bases are adenine (A), thymine (T), guanine (G) and cytosine (C).
- A always pairs with a T, and a C always with a G.

What is sequencing?

- Sequencing means determining the exact order of the base pairs in a segment of DNA.
- Human chromosomes range in size from about 50,000,000 to 300,000,000 bps.
- The genome contains paired strands so the identity of one of the bases in the pair determines the other member of the pair - only one bps from one pair need to be reported.

Sequencing Technologies

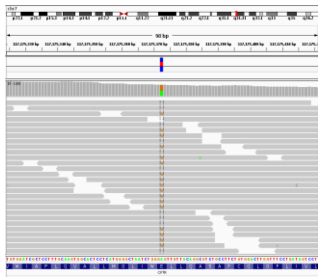
Typical applications of massive (also called next-generation) sequencing technologies:

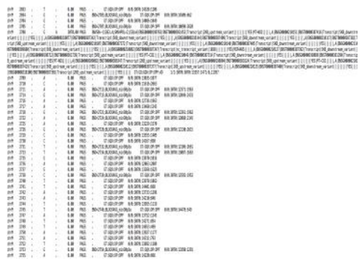
DNA Sequencing

- De novo Genome Assembly
- Single Nucleotide Variation discovery
- Copy Number Variation detection
- Structural Rearrangements

Variant Calling Analysis Pipeline

Tissue collection → Sample preparation → Sequencing → Mapping → Variant Calling





~ 1 day

~ 1/2 days

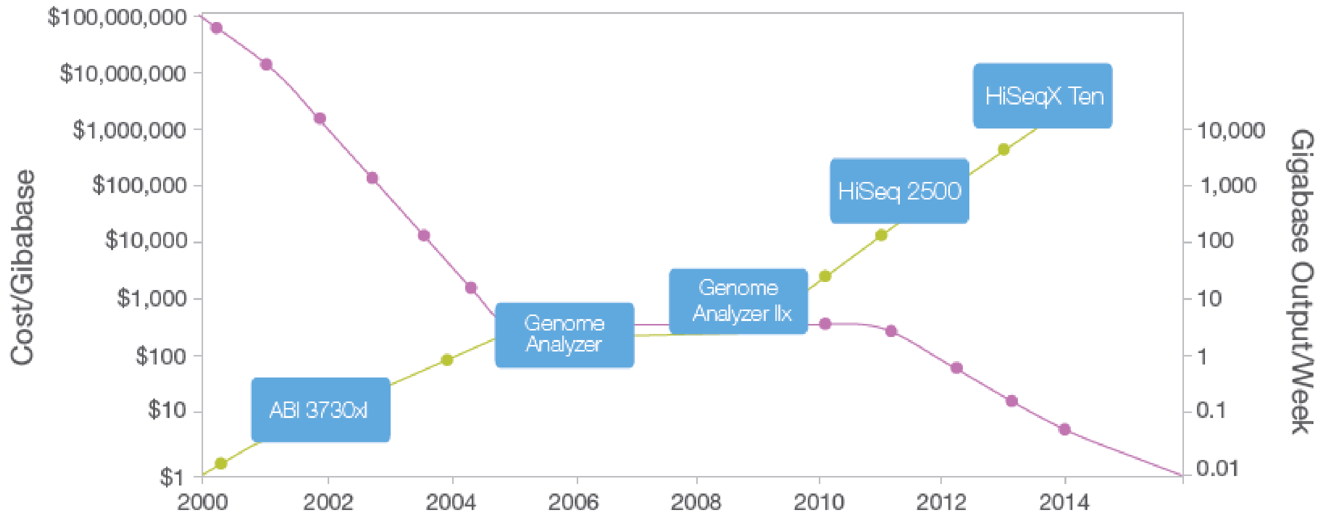
~ 2/5 days

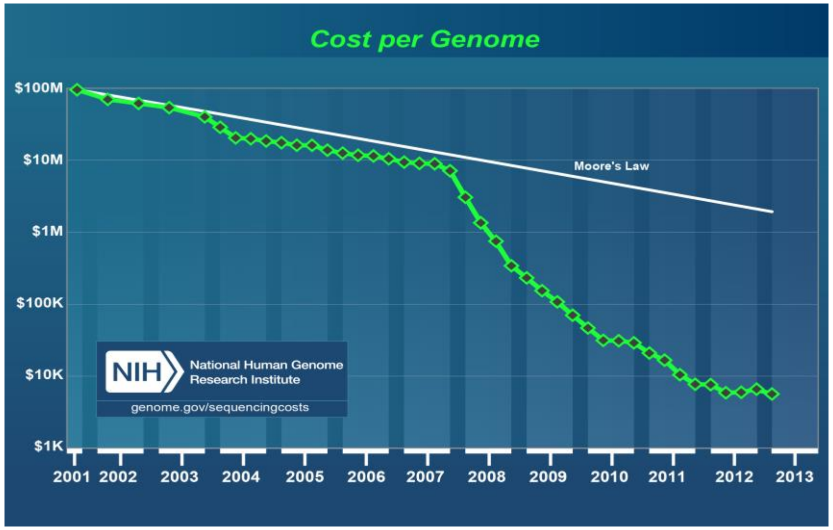
~ 2 days

Several weeks

Sequencing Cost and Data Output

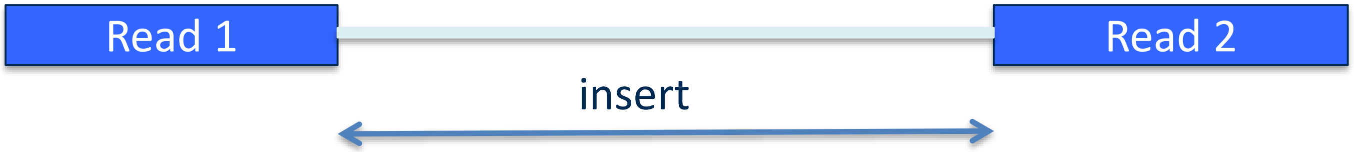
Cost and Ouput of Ilumina sequencing machines across the years

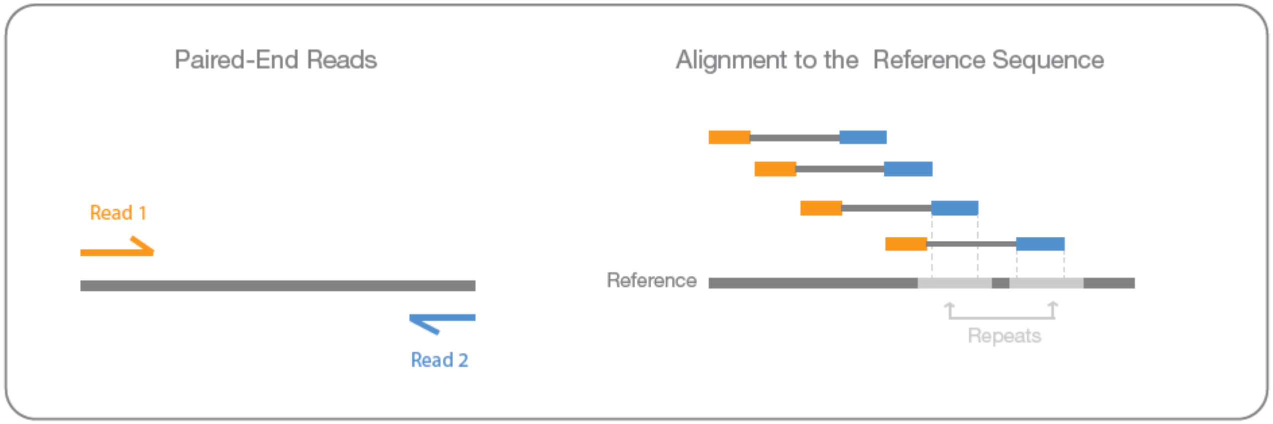




Sequence Reads (Concepts)

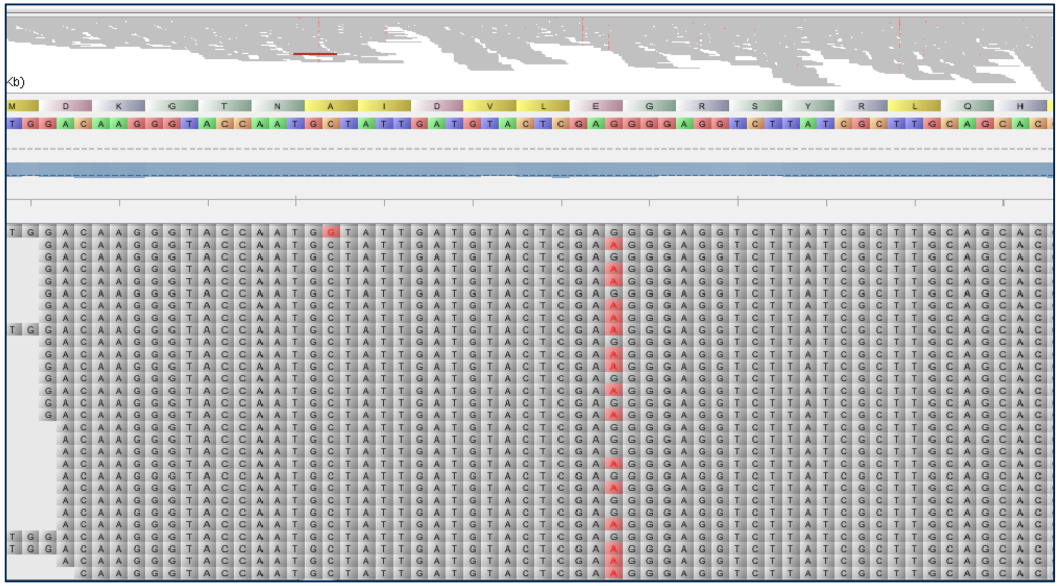
- **Insert** – the DNA fragment that is used for sequencing.
- **Read** – the part of the insert that is sequenced.
- **Single Read (SR)** – a sequencing procedure by which the insert is sequenced from one end only.
- **Paired End (PE)** – a sequencing procedure by which the insert is sequenced from both ends.
- **Insert size** - average distance between read pair mates





Adapted from Illumina.com

Next Generation Sequencing File Formats



Algoritmos Avançados de Bioinformática

File Naming Convention

- Certain programs have strict naming conventions and besides the data format they use information in the filename to interpret aspects of the data.

- File naming conventions make easier the managing of the files.

- Typically a suffix is used to denote the data type.

Full description of several data formats can be found here:
https://genome.ucsc.edu/FAQ/FAQformat.html#format5.1

Most common file formats for next-generation sequencing data:
FASTA, FASTQ, SAM/BAM

other common formats:

BED, WIG, GFF/GTF, VCF

FASTA

FASTA (.fasta, .fa, .fsa)
Possibly the most common biological sequence format. It may contain nucleotide or peptide sequence(s) and a single-line header per sequence.

It contains a header line that starts with ">" followed by the sequenced identifier and description. The next lines contain the sequence.

Format:

Line 1:	Single line description, marked by “>”
Line 2…n:	Lines of sequence data (no longer than 60 characters)

```
$ cd Motifs
$ head tf4.fasta.txt

>gi|557079417|gb|ESQ25234.1| Transcription initiation factor TFIIIB, Brf1
subunit/Transcription initiation factor TFIIB [uncultured Acidilobus sp. OSP8]
MRCPVCGSARLAWDGSTGYLVCQDCGAVISQLIEEERPALPRPPWRPIRRRPEPVVPPASRPLEEDVEEAAAKAVRRGRV
IEIVGRAVRLRPPMKEKVEGLEGLLEMMSGFPDLKSRTERVRKALALYAALRAMGLSRSRATELASRATGASPRSILKVR
ERHQRSLDMYEVAVAEALRQKRLSPPMLAEKLKAGLGPSVHS
```

FASTQ

```
$ cd NGS
$ head top_1.fq
```

~~~
@K00136:112:H3LFLBBXX:5:1102:32191:34473 1:N:0
TGCCAGGGCCCCTGCCTGTGGGACTCCACCCAAAGCAGGTTCACACCAGGACACACCTGGCACCTTTGGCATTGCCTCCCTGTGCC
CTAAGCAATCTCCC
+
```eejje[eejjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjejjjj`jjjjj`jjejjjjjjjjjjjjjjjjjjjjjejj
jjj`ejj``jjjej
~~~

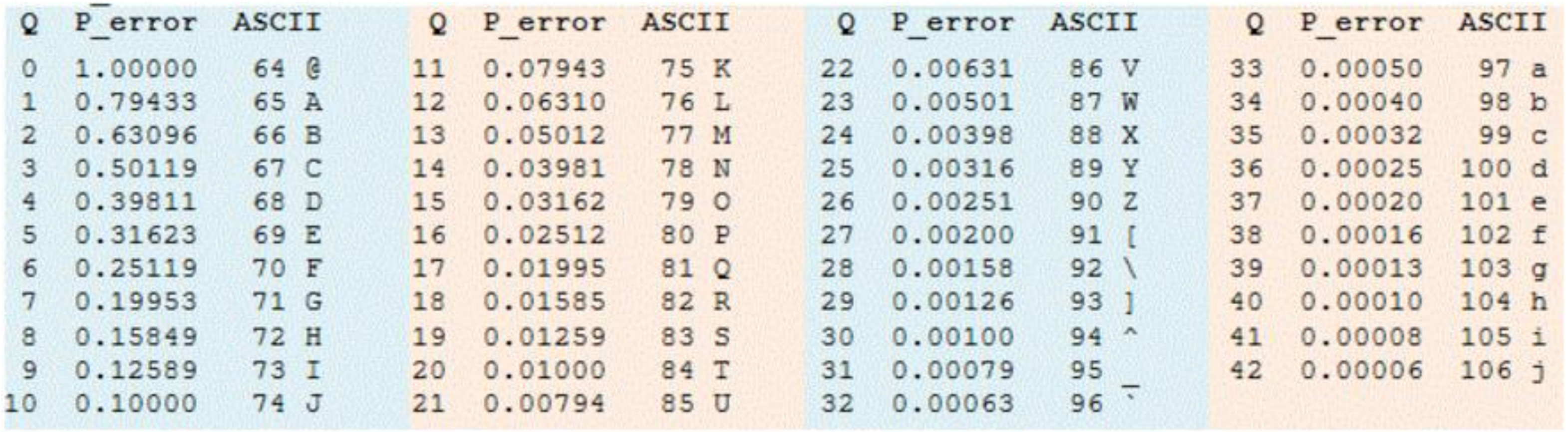
Sequence identifier

Sequence

Quality Score

- For each base pair sequence there is an associated quality score;
- This corresponds to an integer value that represents the probability of an error;
- The score is known as Phred or Q score;
- Each value is encoded as a ASCII character;
- **Q = –10 log10(P)**

Adapted from http://drive5.com/usearch/manual/quality_score.html



Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII
0	1.00000	64 @	11	0.07943	75 K	22	0.00631	86 V	33	0.00050	97 a
1	0.79433	65 A	12	0.06310	76 L	23	0.00501	87 W	34	0.00040	98 b
2	0.63096	66 B	13	0.05012	77 M	24	0.00398	88 X	35	0.00032	99 c
3	0.50119	67 C	14	0.03981	78 N	25	0.00316	89 Y	36	0.00025	100 d
4	0.39811	68 D	15	0.03162	79 O	26	0.00251	90 Z	37	0.00020	101 e
5	0.31623	69 E	16	0.02512	80 P	27	0.00200	91 [	38	0.00016	102 f
6	0.25119	70 F	17	0.01995	81 Q	28	0.00158	92 \	39	0.00013	103 g
7	0.19953	71 G	18	0.01585	82 R	29	0.00126	93]	40	0.00010	104 h
8	0.15849	72 H	19	0.01259	83 S	30	0.00100	94 ^	41	0.00008	105 i
9	0.12589	73 I	20	0.01000	84 T	31	0.00079	95 _	42	0.00006	106 j
10	0.10000	74 J	21	0.00794	85 U	32	0.00063	96 `			

Quality Control of FASTQ files

Use fastQC for QC statistics and a quick graphical overview of the dataset.

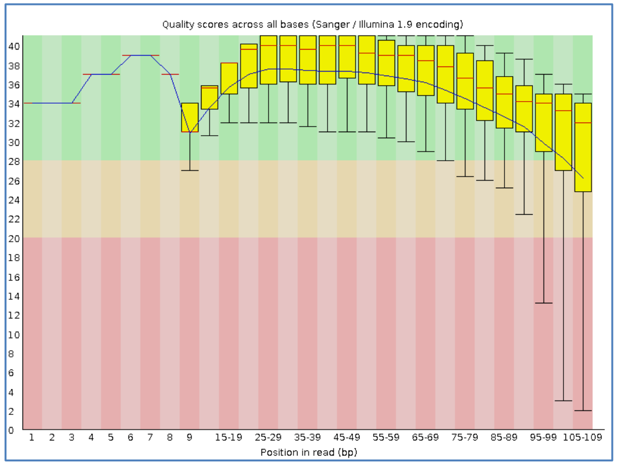
```
# uncompress the assembly taster file
$ tar -xvf assembly_taster.tar.xz
$ cd assembly_taster/

# Run FastQC on the dataset
$ fastqc mt_reads.fastq

# change to mt_reads_fastqc folder
# open report.html in browser by double clicking
```

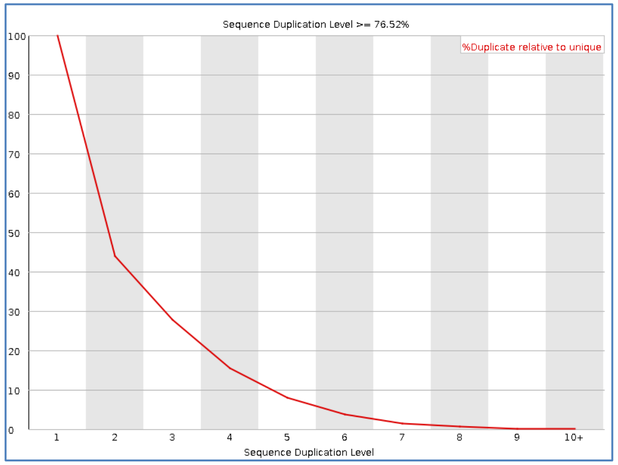
Quality Control of FASTQ files

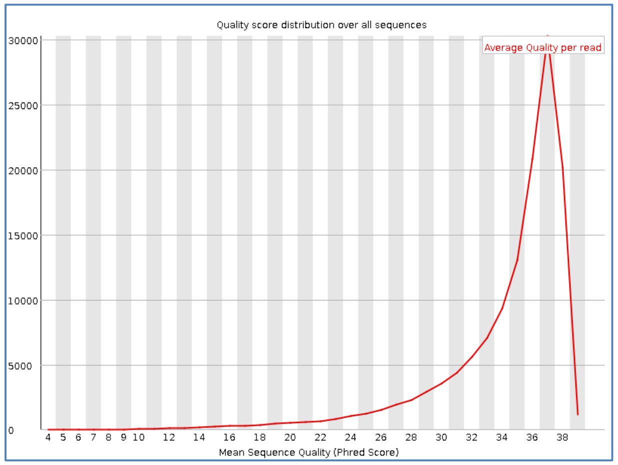
FastQC Report

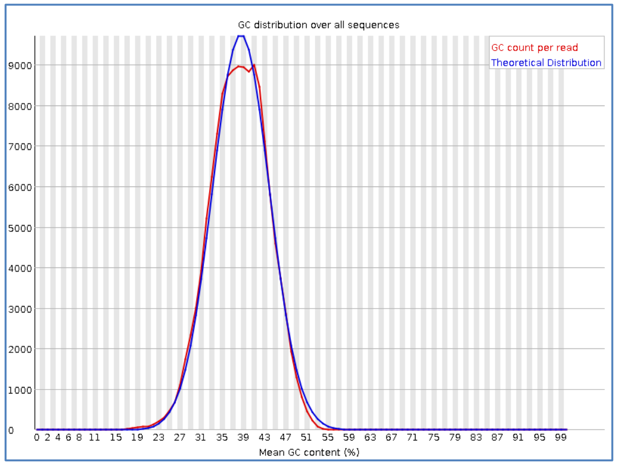
Summary

- ✓ Basic Statistics
- ✓ Per base sequence quality
- ✓ Per sequence quality scores
- ✗ Per base sequence content
- ✗ Per base GC content
- ✓ Per sequence GC content
- ✓ Per base N content
- ✓ Sequence Length Distribution
- ✗ Sequence Duplication Levels
- ✓ Overrepresented sequences
- ✗ Kmer Content









https://www.youtube.com/watch?v=bz93ReOv87Y

SAM/BAM

Sequence Alignment/Map **SAM/BAM (.sam, .bam)** is a tab-delimited file format. SAM files contain detailed and rich mapping information of the reads against the reference genome/sequence. SAM is a text format and BAM is a binary version of SAM for efficient indexing.

It contains an header lines with meta information that start with the '@' symbol. Each alignment line consists of following fields:

Format:

1. QNAME Query template/pair NAME
2. FLAG bitwise FLAG
3. RNAME Reference sequence NAME
4. POS 1-based leftmost POSition/coordinate of clipped sequence
5. MAPQ MAPping Quality (Phred-scaled)
6. CIGAR extended CIGAR string
7. MRNM Mate Reference sequence NaMe ('=' if same as RNAME)
8. MPOS 1-based Mate POSistion
9. LEN inferred Template LENgth (insert size)
10. SEQ query SEQuence on the same strand as the reference
11. QUAL query QUALity (ASCII-33 gives the Phred base quality)
12. OPT variable OPTional fields in the format TAG:VTYPE:VALUE

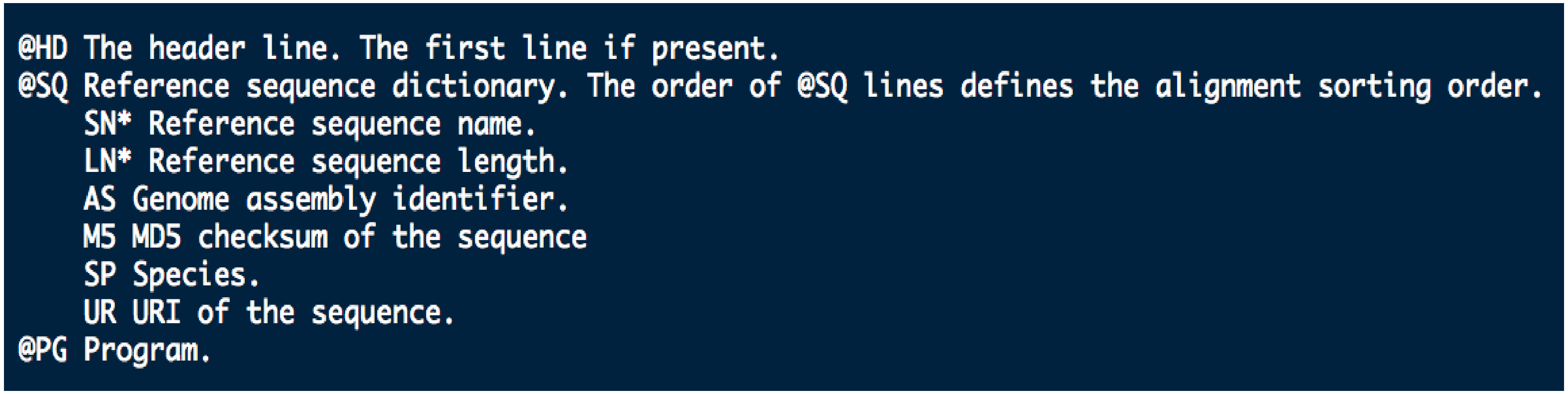
SAM/BAM

Samtools is a package of tools to efficiently manipulate SAM and BAM files.

```
Usage:    samtools <command> [options]

Command: view           SAM<->BAM conversion
         sort           sort alignment file
         mpileup        multi-way pileup
         depth          compute the depth
         faidx          index/extract FASTA
         tview          text alignment viewer
         index          index alignment
         idxstats       BAM index stats (r595 or later)
         fixmate        fix mate information
         flagstat       simple stats
         calmd          recalculate MD/NM tags and '=' bases
         merge          merge sorted alignments
         rmdup          remove PCR duplicates
         reheader       replace BAM header
         cat            concatenate BAMs
         bedcov         read depth per BED region
         targetcut      cut fosmid regions (for fosmid pool only)
         phase          phase heterozygotes
         bamshuf        shuffle and group alignments by name
```

SAM/BAM

SAM files have an header (optional) that provide metadata about the alignment.



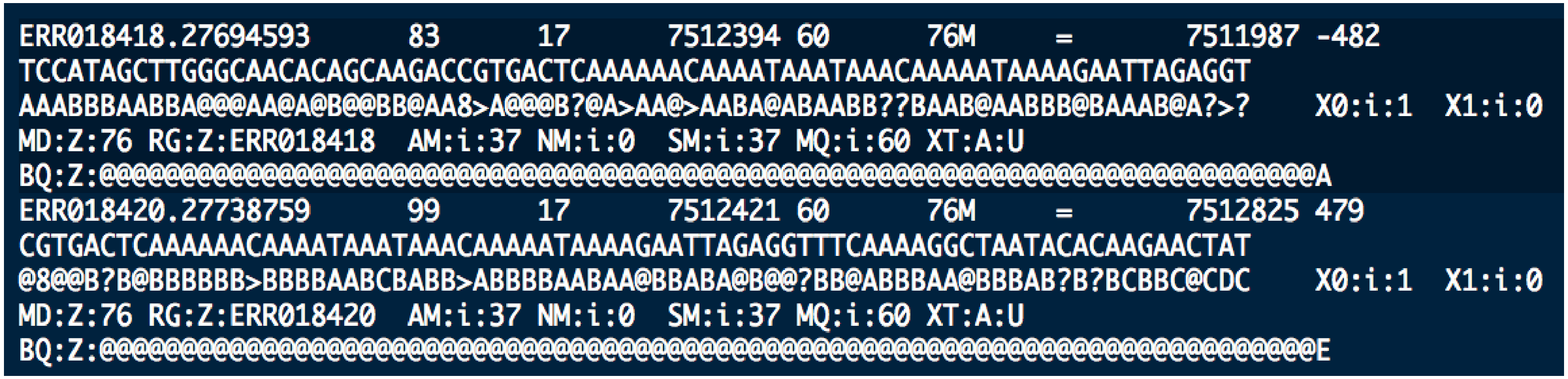
```
@HD The header line. The first line if present.
@SQ Reference sequence dictionary. The order of @SQ lines defines the alignment sorting order.
    SN* Reference sequence name.
    LN* Reference sequence length.
    AS Genome assembly identifier.
    M5 MD5 checksum of the sequence
    SP Species.
    UR URI of the sequence.
@PG Program.
```

SAM/BAM

Samtools is a package of tools to efficiently manipulate SAM and BAM files.

```
$ samtools view -h
ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase1/data/HG00154/alignment/HG0015
4.mapped.ILLUMINA.bwa.GBR.low_coverage.20101123.bam 17:7512445-7513455 >
example.sam

$ less example.sam
```

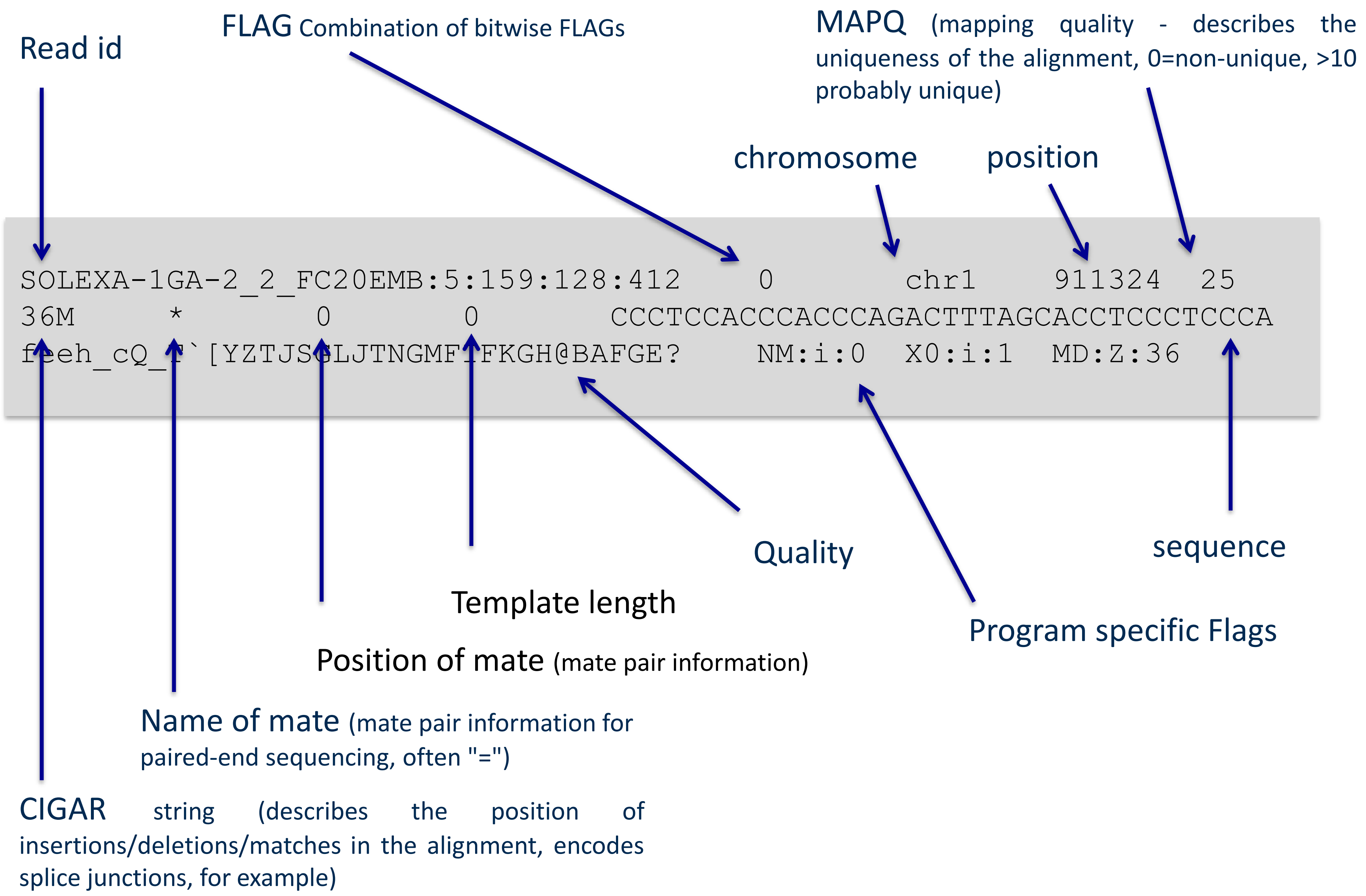
Each row defines a single raw read alignment against the reference genome.



```
ERR018418.27694593      83      17      7512394 60      76M     =       7511987 -482
TCCATAGCTTGGGCAACACAGCAAGACCGTGACTCAAAAAACAAAATAAATAAACAAAAATAAAAGAATTAGAGGT
AAABBBAABBA@@@AA@A@B@@BB@AA8>A@@@B?@A>AA@>AABA@ABAABB??BAAB@AABBB@BAAAB@A?>?    X0:i:1  X1:i:0
MD:Z:76 RG:Z:ERR018418  AM:i:37 NM:i:0  SM:i:37 MQ:i:60 XT:A:U
BQ:Z:@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@A
ERR018420.27738759      99      17      7512421 60      76M     =       7512825 479
CGTGACTCAAAAAACAAAATAAATAAACAAAAATAAAAGAATTAGAGGTTTCAAAAGGCTAATACACAAGAACTAT
@8@@B?B@BBBBBB>BBBBAABCBABB>ABBBBAABAA@BBABA@B@@?BB@ABBBAA@BBBAB?B?BCBBC@CDC    X0:i:1  X1:i:0
MD:Z:76 RG:Z:ERR018420  AM:i:37 NM:i:0  SM:i:37 MQ:i:60 XT:A:U
BQ:Z:@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@E
```

For further details:
"*The Sequence Alignment/Map format and SAMtools*", Heng Li et al, Bioinformatics 2009.

SAM/BAM



Read id

FLAG Combination of bitwise FLAGs

MAPQ (mapping quality - describes the uniqueness of the alignment, 0=non-unique, >10 probably unique)

chromosome

position

```
SOLEXA-1GA-2_2_FC20EMB:5:159:128:412     0         chr1     911324   25
36M       *          0         0         CCCTCCACCCACCCAGACTTTAGCACCTCCCTCCCA
feeh_cQ_T`[YZTJSGLJTNGMFTFKGH@BAFGE?     NM:i:0   X0:i:1   MD:Z:36
```

Quality

sequence

Template length

Program specific Flags

Position of mate (mate pair information)

Name of mate (mate pair information for paired-end sequencing, often "=")

CIGAR string (describes the position of insertions/deletions/matches in the alignment, encodes splice junctions, for example)

SAM/BAM

Manipulate a BAM file, by extracting some alignments from it and create a new BAM file.

Get an example RNA-seq BAM file from ENCODE project.

```
$ wget
http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeUwRe
pliSeq/wgEncodeUwRepliSeqBg02esG1bAlnRep1.bam

$ samtools view -h wgEncodeUwRepliSeqBg02esG1bAlnRep1.bam | head -
1000  > top.sam

$ samtools view -b top.sam > top.bam
```

BED

BED (Browser Extensible Data) is flexible data format that defines the genomic coordinates of defined features (exons, genes, ESTs, ...). Each line defines the coordinates of the feature and there are three required fields (BED3) and nine additional optional fields. The number of fields per line must be consistent throughout the file.

The first three required BED fields are:

1. chrom - The name of the chromosome (e.g. chr3, chrY, chr2_random).
2. chromStart - The starting position of the feature in the chromosome.
3. chromEnd - The ending position of the feature in the chromosome.

Other three common fields:

4. name - Defines the name of the feature.
5. score - A score between 0 and 1000 for coloring purposes.
6. strand - Defines the strand - either '+' or '-'.

```
$ cd GenomeAnnotation
$ head gene.regions.bed
# Another example file
track name=pairedReads description="Clone Paired Reads" useScore=1
chr22 1000 5000 cloneA 960 + 1000 5000 0 2 567,488, 0,3512
chr22 2000 6000 cloneB 900 - 2000 6000 0 2 433,399, 0,3601
```

WIG

Wiggle format (**WIG**) allows the display of continuous-valued data in a track format. Wiggle format is line-oriented. It defines genomic intervals/regions with an associated value to be plotted.

- There are two options for formatting wiggle data: *variableStep* and *fixedStep*.

```
# variableStep is for data with irregular intervals between new data
points.
# Example
variableStep chrom=chr2
300701 12.5
300801 15.0
300903 10.5
300940 8.9
```

```
# fixedStep is for data with regular intervals between new data
values and is the more compact wiggle format.
# Example
fixedStep chrom=chr3 start=400601 step=100
11
22
33
```

GFF/GTF

GFF (General Feature Format) is a format used to describe gene structure annotation. GFF lines have nine required fields that must be tab-separated.

Format:

1. seqname - The name of the sequence. Must be a chromosome or scaffold.
2. source - The program that generated this feature.
3. feature - The name of this type of feature. Some examples of standard feature types are "CDS", "start_codon", "stop_codon", and "exon".
4. start - The starting position of the feature in the sequence. The first base is numbered 1.
5. end - The ending position of the feature (inclusive).
6. score - A score between 0 and 1000. If the track line useScore attribute is set to 1 for this annotation data set, the score value will determine the level of gray in which this feature is displayed (higher numbers = darker gray). If there is no score value, enter ".".
7. strand - Valid entries include '+', '-', or '.' (for don't know/don't care).
8. frame - If the feature is a coding exon, frame should be a number between 0-2 that represents the reading frame of the first base. If the feature is not a coding exon, the value should be '.'.
9. group - All lines with the same group are linked together into a single item.

GFF/GTF

GTF (Gene Transfer Format) is a refinement to GFF that tightens the specification. The first eight GTF fields are the same as GFF. The group field has been expanded into a list of attributes.

```
$ cd GenomeAnnotation
$ gzcat genes.gtf.zip | head

1  unknown  exon    11874   12227    .    +    .    gene_id "LOC100287102";
   gene_name "LOC100287102"; p_id "P25115"; transcript_id
   "XM_002342010.1"; tss_id "TSS26210";

1   unknown CDS 12190   12227    .    +    0    gene_id "LOC100287102";
gene_name "LOC100287102"; p_id "P25115"; transcript_id
"XM_002342010.1"; tss_id "TSS26210";
```

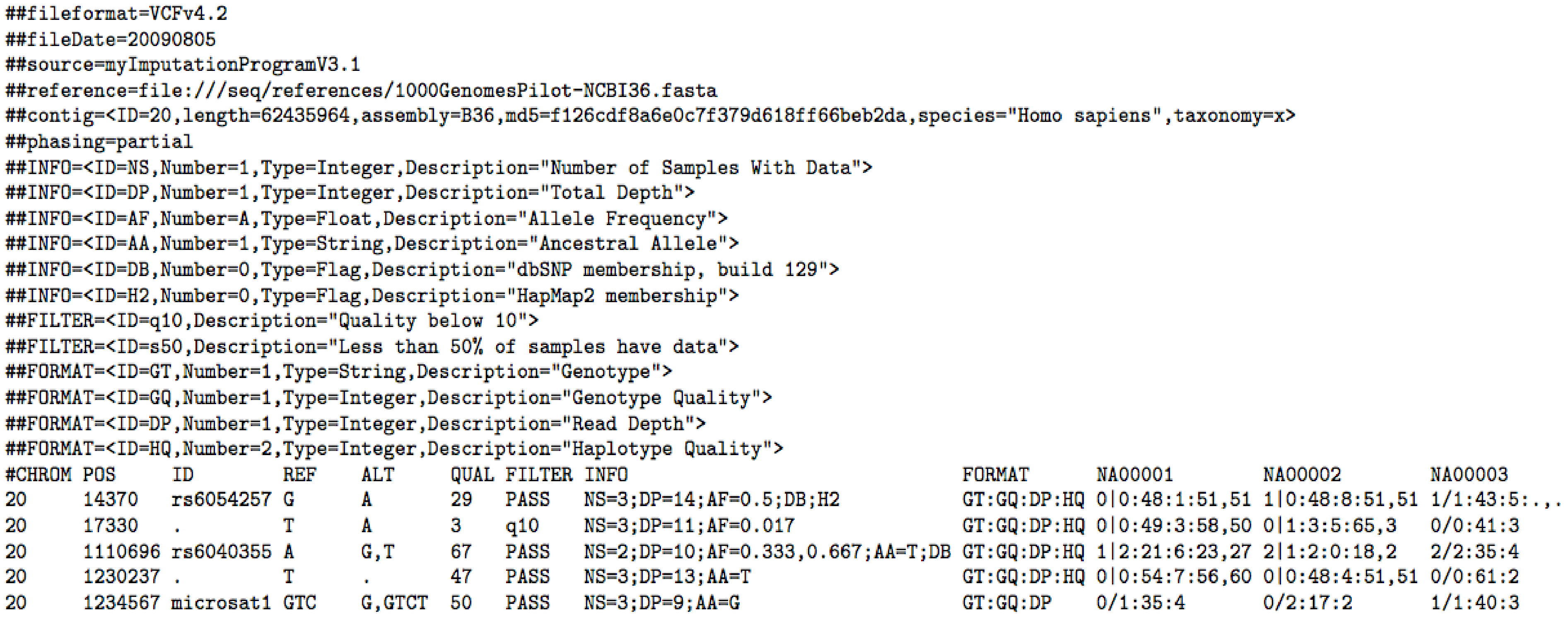
VCF

Variant Call Format (**VCF**) is a flexible and extendable format for variation data such as single nucleotide variants, insertions/deletions, copy number variants and structural variants. This format is defined by the 1000 Genomes consortium: http://www.1000genomes.org/wiki/Analysis/Variant%20Call%20Format/vcf-variant-call-format-version-41 and the GA4GH working group: http://ga4gh.org/#/fileformats-team

There are 8 fixed fields per record. All data lines are tab-delimited.

1. CHROM - chromosome:
2. POS - position: The reference position
3. ID - identifier: Semi-colon separated list of unique identifiers where available. If this is a dbSNP variant it is encouraged to use the rs number(s).
4. REF - reference base(s): Each base must be one of A,C,G,T,N (case insensitive). Multiple bases are permitted.
5. ALT - alternate base(s): Comma separated list of alternate non-reference (A,C,G,T,N,*) alleles called on at least one of the samples.
6. QUAL - quality: Phred-scaled quality score for the assertion made in ALT. i.e. ?10log10 prob(call in ALT is wrong).
7. FILTER - filter status: PASS if this position has passed all filters, i.e. a call is made at this position.
8. INFO - additional information: INFO fields are encoded as a semicolon-separated series of short keys with optional values in the format: <key>=<data>[,data].

VCF Example



```
##fileformat=VCFv4.2
##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
```

Example of a VCF file. Adapted from the VCF specification document: http://samtools.github.io/hts-specs/VCFv4.2.pdf

VCF Example

Example files for testing are provided by Illumina. Whole-genome sequencing performed on Illumina HiSeq® systems is enabling researchers worldwide to more fully and accurately characterize the human genome.

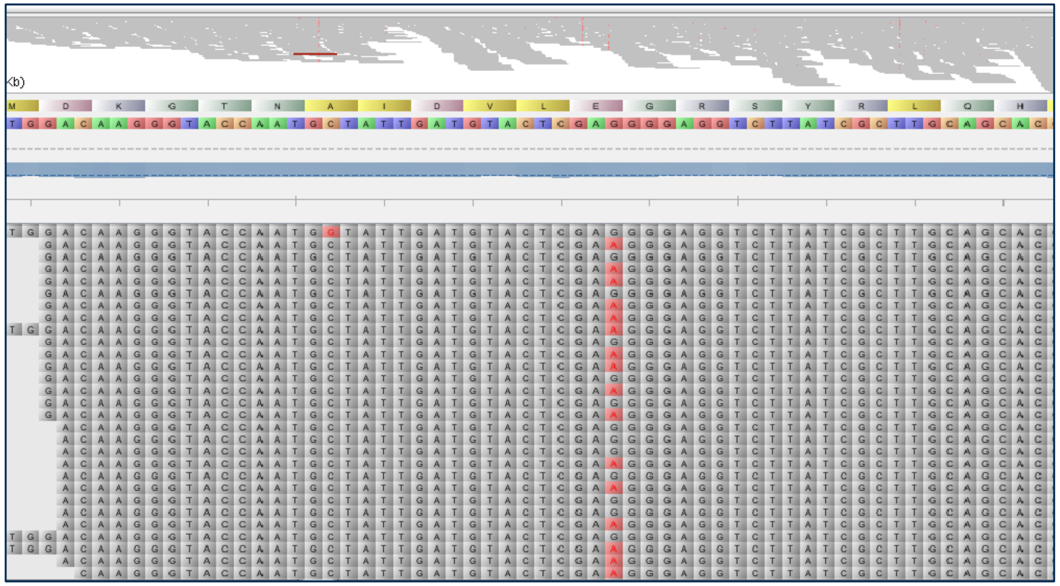
Illumina have derived a set of high-confidence variant calls for NA12877 and NA12878 (part of the 1000 Genomes Project). The results of the analysis, VCF files and documentation, can be obtained from here:
ftp://ussd-ftp.illumina.com/

Example vcf file for NA12878:

```
$  wget ftp://platgene_ro@ussd-ftp.illumina.com/hg19/8.0.1/NA12878/NA12878.vcf.gz
```

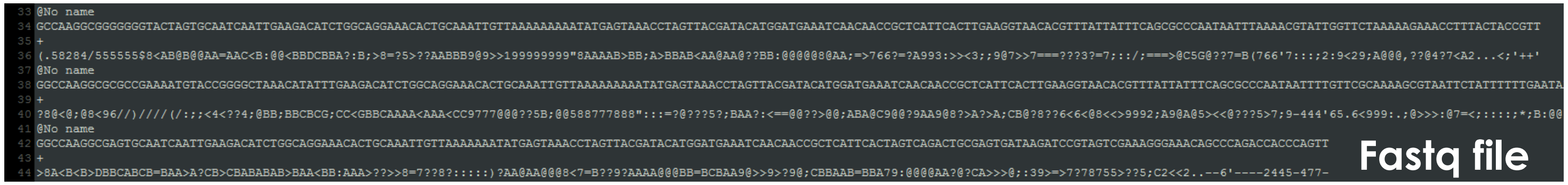
```
$ wget
ftp-
trace.ncbi.nih.gov/1000genomes/ftp/release/20110521/ALL.chr17.phase1_re
lease_v3.20101123.snps_indels_svs.genotypes.vcf.gz
```

Variant Calling Analysis



Algoritmos Avançados de Bioinformática

Alignment and Variant Calling (Bioinformatics Analysis)



```
@No name
GCCAAGGCGGGGGGGTACTAGTGCAATCAATTGAAGACATCTGGCAGGAAACACTGCAAATTGTTAAAAAAAAATATGAGTAAACCTAGTTACGATACATGGATGAAATCAACAACCGCTCATTCACTTGAAGGTAACACGTTTATTATTTCAGCGCCCAATAATTTAAAACGTATTGGTTCTAAAAAGAAACCTTTACTACCGTT
+
(.58284/555555$8<AB@B@@AA=AAC<B:@@<BBDCBBA?:B;>8=?5>??AABBB9@9>>199999999"8AAAAB>BB;A>BBAB<AA@AA@??BB:@@@@@8@AA;=>766?=?A993:>><3;;9@7>>7===???3?=7;::/;===>@C5G@??7=B(766'7:::;2:9<29;A@@@,??@4?7<A2...<;'++'
@No name
GGCCAAGGCGCGCCGAAAATGTACCGGGGCTAAACATATTTGAAGACATCTGGCAGGAAACACTGCAAATTGTTAAAAAAAAAATATGAGTAAACCTAGTTACGATACATGGATGAAATCAACAACCGCTCATTCACTTGAAGGTAACACGTTTATTATTTCAGCGCCCAATAATTTTGTTCGCAAAAGCGTAATTCTATTTTTTGAATA
+
?8@<@;@8<96//)////(/:;;<4<??4;@BB;BBCBCG;CC<GBBCAAAA<AAA<CC9777@@@??5B;@@588777888":::=?@???5?;BAA?:<==@@??>@@;ABA@C9@@?9AA9@8?>A?>A;CB@?8??6<6<@8<<>9992;A9@A@5><<@???5>7;9-444'65.6<999:.;@>>>:@7=<;::::;*;B:@@
@No name
GGCCAAGGCGAGTGCAATCAATTGAAGACATCTGGCAGGAAACACTGCAAATTGTTAAAAAAAATATGAGTAAACCTAGTTACGATACATGGATGAAATCAACAACCGCTCATTCACTAGTCAGACTGCGAGTGATAAGATCCGTAGTCGAAAGGGAAACAGCCCAGACCACCCAGTT
+
>8A<B<B>DBBCABCB=BAA>A?CB>CBABABAB>BAA<BB:AAA>??>>8=7??8?::::)?AA@AA@@@8<7=B??9?AAAA@@@BB=BCBAA9@>>9>?9@;CBBAAB=BBA79:@@@@AA?@?CA>>>@;:39>=>7?78755>??5;C2<<2..--6'----2445-477-
```

Fastq file

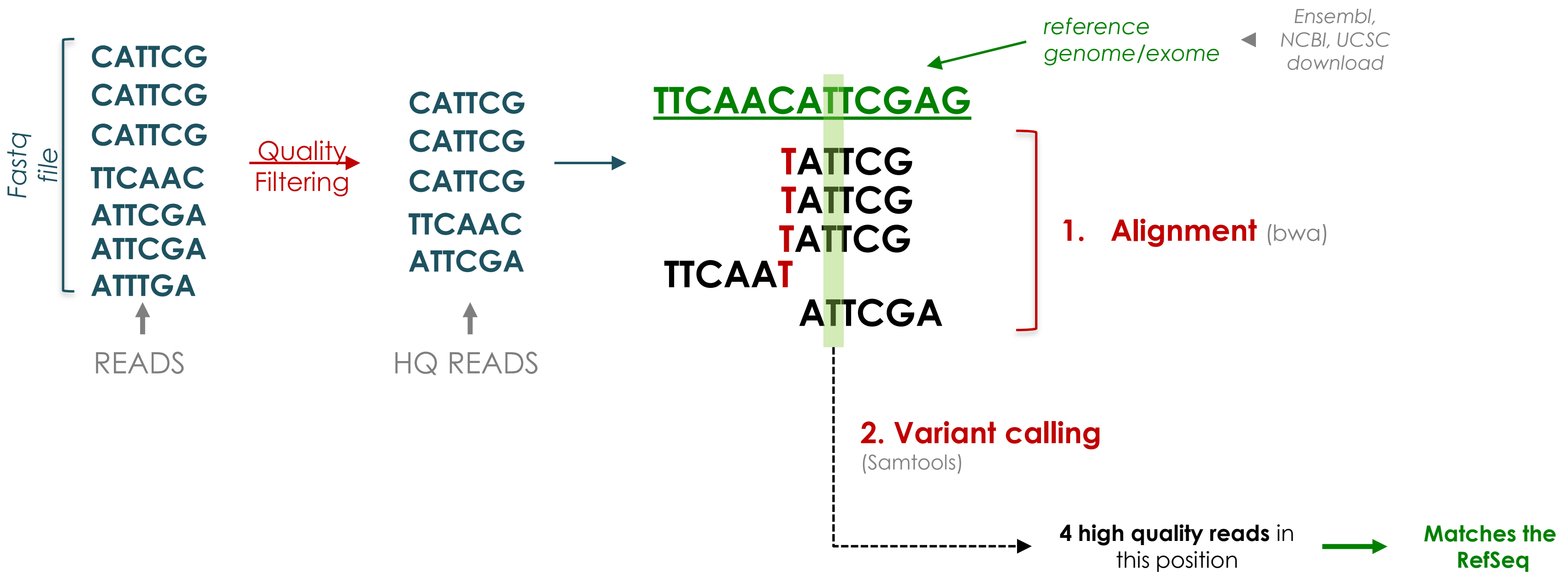
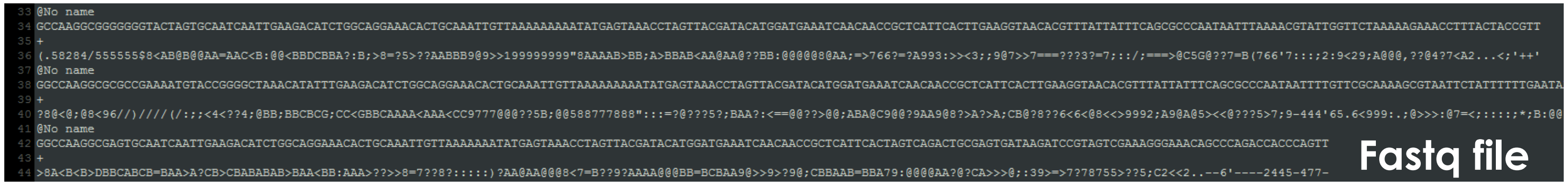


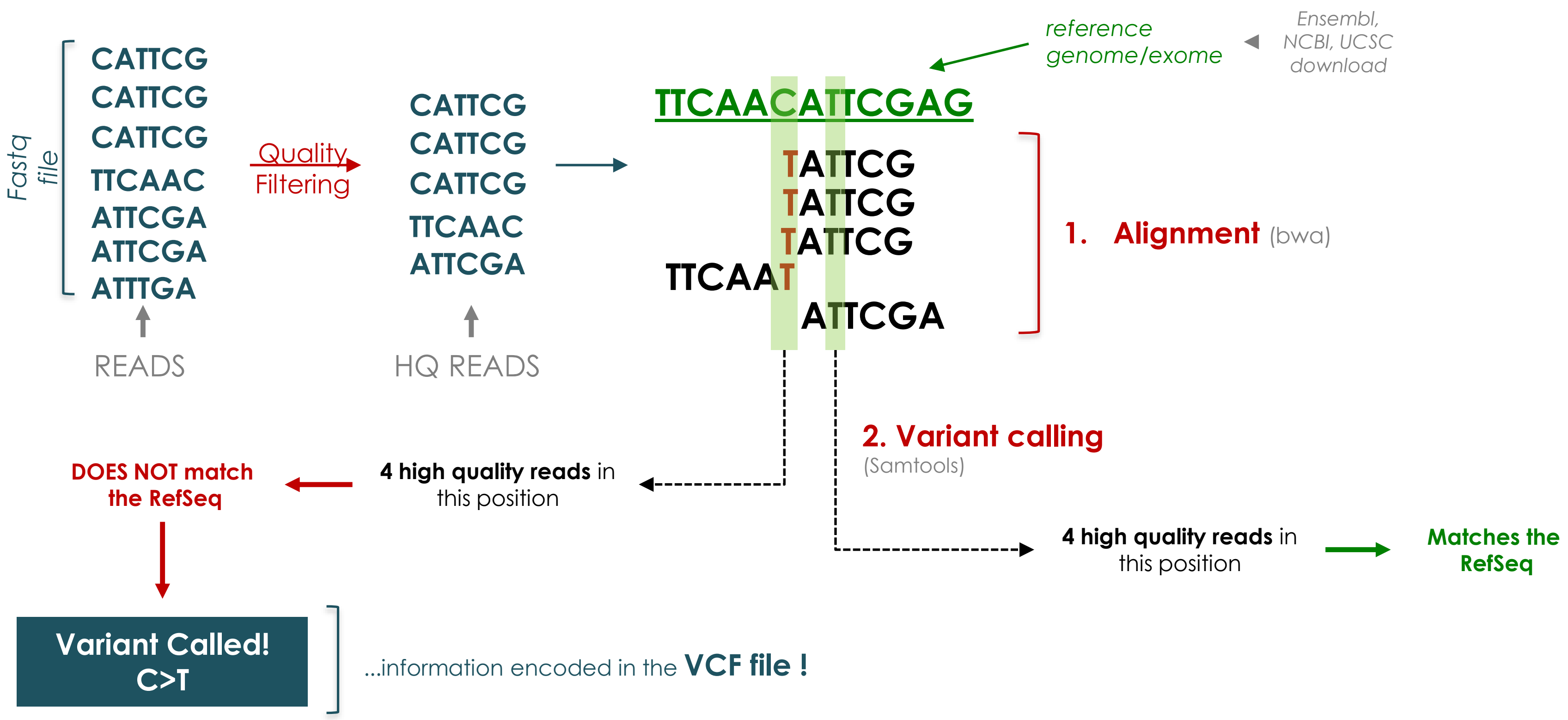
Figure adapted from Patricia Oliveira

Alignment and Variant Calling (Bioinformatics Analysis)

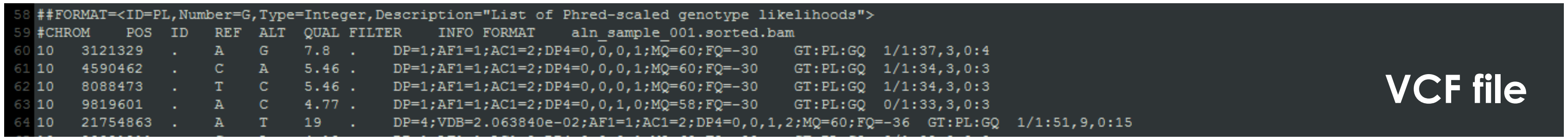


```
@No name
GCCAAGGCGGGGGGGTACTAGTGCAATCAATTGAAGACATCTGGCAGGAAACACTGCAAATTGTTAAAAAAAAAATATGAGTAAACCTAGTTACGATACATGGATGAAATCAACAACCGCTCATTCACTTGAAGGTAACACGTTTATTATTTCAGCGCCCAATAATTTAAAACGTATTGGTTCTAAAAAGAAACCTTTACTACCGTT
+
(.58284/555555$8<AB@B@@AA=AAC<B:@@<BBDCBBA?:B;>8=?5>??AABBB9@9>>199999999"8AAAAB>BB;A>BBAB<AA@AA@??BB:@@@@@8@AA;=>766?=?A993:>><3;;9@7>>7===???3?=7;::/;===>@C5G@??7=B(766'7:::;2:9<29;A@@@,??@4?7<A2...<;'++'
@No name
GGCCAAGGCGCGCCGAAAATGTACCGGGGCTAAACATATTTGAAGACATCTGGCAGGAAACACTGCAAATTGTTAAAAAAAAAATATGAGTAAACCTAGTTACGATACATGGATGAAATCAACAACCGCTCATTCACTTGAAGGTAACACGTTTATTATTTCAGCGCCCAATAATTTTGTTCGCAAAAGCGTAATTCTATTTTTTGAATA
+
?8@<@;@8<96//)////(/:;;<4<??4;@BB;BBCBCG;CC<GBBCAAAA<AAA<CC9777@@@??5B;@@588777888":::=?@???5?;BAA?:<==@@??>@@;ABA@C9@@?9AA9@8?>A?>A;CB@?8??6<6<@8<<>9992;A9@A@5><<@???5>7;9-444'65.6<999:.;@>>>:@7=<;::::;*;B:@@
@No name
GGCCAAGGCGAGTGCAATCAATTGAAGACATCTGGCAGGAAACACTGCAAATTGTTAAAAAAAATATGAGTAAACCTAGTTACGATACATGGATGAAATCAACAACCGCTCATTCACTAGTCAGACTGCGAGTGATAAGATCCGTAGTCGAAAGGGAAACAGCCCAGACCACCCAGTT
+
>8A<B<B>DBBCABCB=BAA>A?CB>CBABABAB>BAA<BB:AAA>??>>8=7??8?:::::)?AA@AA@@@8<7=B??9?AAAA@@@BB=BCBAA9@>>9>?9@;CBBAAB=BBA79:@@@@AA?@?CA>>>@;:39>=>7?78755>??5;C2<<2..--6'----2445-477-
```

Fastq file



...information encoded in the **VCF file !**



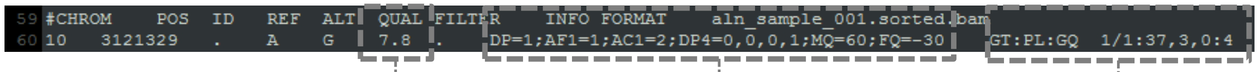
```
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="List of Phred-scaled genotype likelihoods">
#CHROM    POS   ID   REF  ALT  QUAL FILTER    INFO FORMAT    aln_sample_001.sorted.bam
10   3121329   .    A    G    7.8  .    DP=1;AF1=1;AC1=2;DP4=0,0,0,1;MQ=60;FQ=-30    GT:PL:GQ  1/1:37,3,0:4
10   4590462   .    C    A    5.46 .    DP=1;AF1=1;AC1=2;DP4=0,0,0,1;MQ=60;FQ=-30    GT:PL:GQ  1/1:34,3,0:3
10   8088473   .    T    C    5.46 .    DP=1;AF1=1;AC1=2;DP4=0,0,0,1;MQ=60;FQ=-30    GT:PL:GQ  1/1:34,3,0:3
10   9819601   .    A    C    4.77 .    DP=1;AF1=1;AC1=2;DP4=0,0,1,0;MQ=58;FQ=-30    GT:PL:GQ  0/1:33,3,0:3
10   21754863  .    A    T    19   .    DP=4;VDB=2.063840e-02;AF1=1;AC1=2;DP4=0,0,1,2;MQ=60;FQ=-36  GT:PL:GQ  1/1:51,9,0:15
```

VCF file

Figure adapted from Patricia Oliveira

Variant Filtering based on Calling Quality

VCF file version 4.1



```
#CHROM    POS   ID   REF  ALT   QUAL FILTER    INFO FORMAT    aln_sample_001.sorted.bam
10   3121329   .    A    G    7.8  .    DP=1;AF1=1;AC1=2;DP4=0,0,0,1;MQ=60;FQ=-30   GT:PL:GQ  1/1:37,3,0:4
```

QUAL = Quality

Definition: Phred-scaled quality score for the assertion made for the alternative allele i.e. the probability that the call is wrong. Ranges from 1 to 225.

Example: QUAL=225, probability of error is $10^{-22.5}$

High quality scores indicate high confidence calls

Phred Quality Score	Probability of incorrect base call	Base call Accuracy
10	1 in 10	90%
30	1 in 1000	99.90%
40	1 in 10,000	99.99%
50	1 in 100,000	100.00%

INFO = Information

Various information including depth, allele counts, etc

DP=1;AF1=1;AC1=2;DP4=0 ,0,0,1; MQ=60;FQ=-30

Genotype

GT:PL:GQ   1/1:37,3,0:4

GT: Genotype (0/1, 1/1)

PL: Phred-scaled likelihood of genotype
0 is the most likely genotype

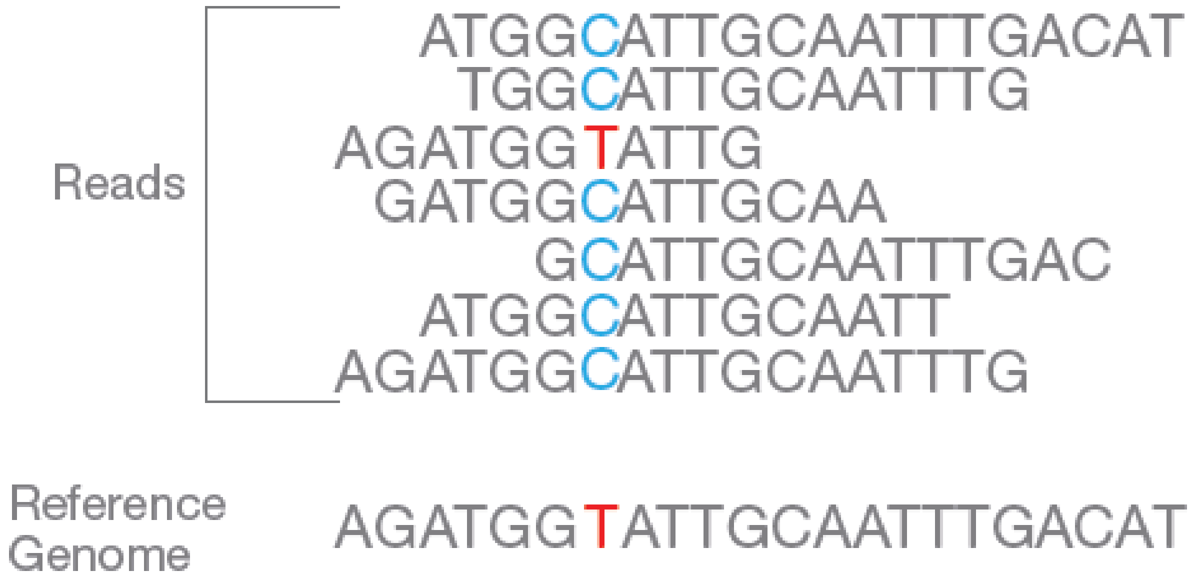
GQ: Phred-scaled quality of genotype
Ranges from 1-99

High genotype quality scores indicate high confidence genotype calls

High confidence variant calls selected!

Adapted from Patricia Oliveira

Single Nucleotide Variant detection



Adapted from Illumina.com

Map reads to the genome (chr21)

We will use a dataset from the 1000 Genomes Project. We will focus on the reads mapping on chr21.

Retrieve the file that contains the sequence from chr21 and the sequencing file with reads mapping on chr21 previously selected:

https://www.dropbox.com/s/jyn94xo4aokdvup/dna.tar.gz?dl=0

bwa : Burrows-Wheeler Alignment Tool. For mapping low-divergent sequences against a large reference genome, such as the human genome.

```
# Build an index for the sequence of chr 21
$ bwa index 21.fa

# align reads to reference sequence, getting genomic coordinates
$ bwa aln 21.fa HG00154.chr21.fastq.gz > HG00154.chr21.sai
```

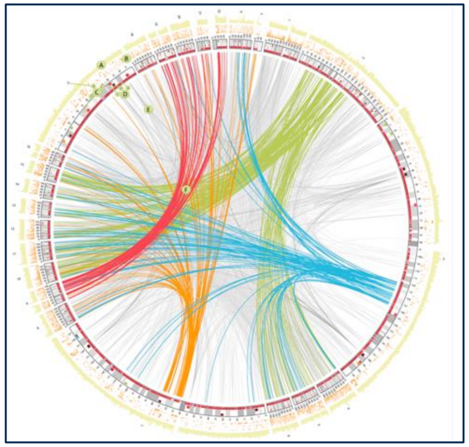
Generate a BAM file with alignments

```
# create a sam file for mapped reads
$  bwa  samse  -f  HG00154.chr21.aln.sam  21.fa  HG00154.chr21.sai
HG00154.chr21.fastq.gz


# convert sam to bam
$ samtools view -S HG00154.chr21.aln.sam -b > HG00154.chr21.aln.bam


# sort by genomic coordinate
$ samtools sort HG00154.chr21.aln.bam HG00154.chr21.aln.sort


# create index for the bam file
$ samtools index HG00154.chr21.aln.sort.bam


# obtain alignment statistics
$ samtools flagstat HG00154.chr21.aln.sort.bam
```

Variant calling

```
#  Variant calling
$  samtools  mpileup  -uf  21.fa  HG00154.chr21.aln.sort.bam  |  bcftools
view -vcg - > HG00154.chr21.raw.vcf

# Inspecting base-level alignments
$ samtools tview HG00154.chr21.aln.sort.bam 21.fa -p 21:9417961
```

High-throughput Sequencing Assembly



This course is adapted from the Bio-Linux Tutorial and Illumina.com documentation

Algoritmos Avançados de Bioinformática

Fragment Assembly in DNA sequencing

Fragment assembly is the task of reconstructing the entire genomic DNA sequence.

Challenges:

- Sequencing errors

  The error rate in DNA reads in current sequencing machines ranges from 1% to 3%.

- Assignment of reads to one of two strands

  DNA is double-stranded, so reads can come from one of two strands

- repeats in DNA (very hard problem)

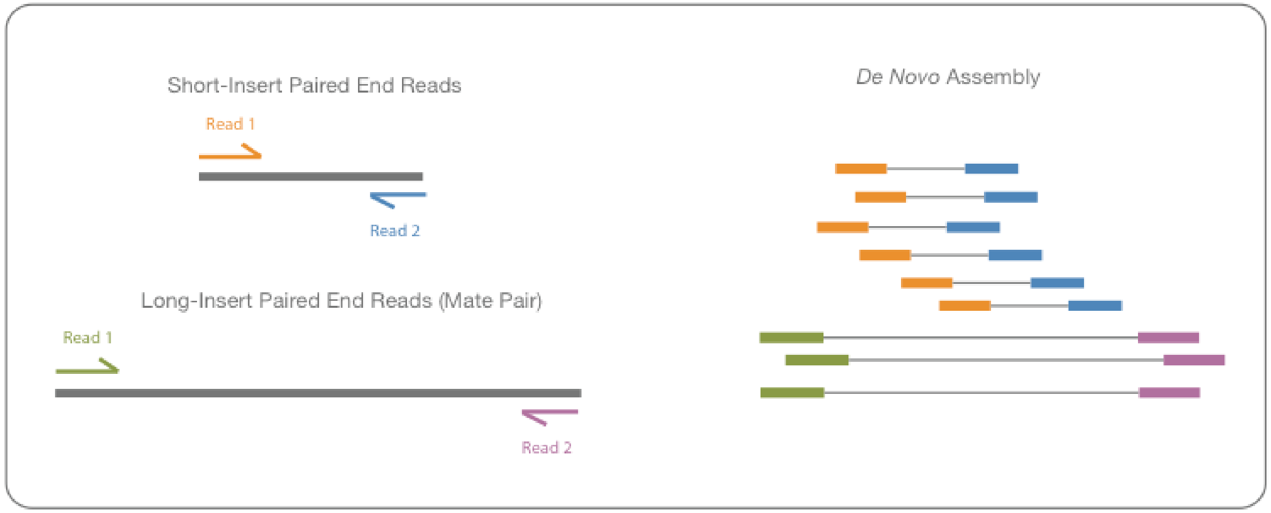
  Depending on the species the genome sequence contains many repeated sequences, which can be as much as 50% of the entire genomes. Besides that it has many genes with duplicated sequences.

Assembly steps

Most of the fragment assembly algorithms consist of the following three steps:

1) Overlap

Finding the potential overlapping reads. Due to sequencing errors a variation of a DP algorithm for this step;


Short-Insert Paired End Reads
Read 1
Read 2
Long-Insert Paired End Reads (Mate Pair)
Read 1
Read 2
De Novo Assembly

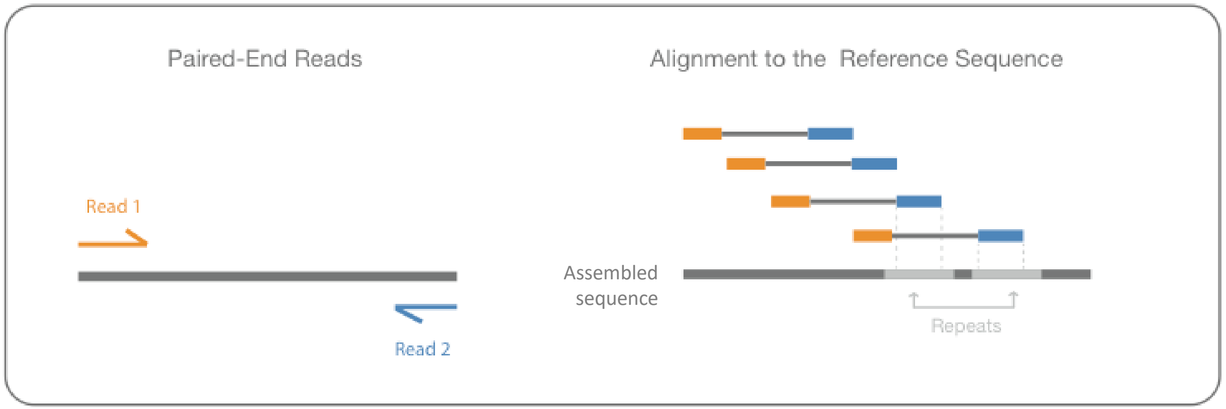
Adapted from Illumina.com

Assembly steps

Most of the fragment assembly algorithms consist of the following three steps:

2) Layout

Constructing the layout consists in deciding if reads actually overlap (and their difference is due to sequencing errors) or they originate from repeat regions (repeats make layout problem very difficult). Mate pair reads help to alleviate this problem.



Adapted from Illumina.com

Assembly steps

Most of the fragment assembly algorithms consist of the following three steps:

1) Overlap

Finding the potential overlapping reads. Due to sequencing errors a variation of a DP algorithm for this step;

2) Layout

Constructing the layout consists in deciding if reads actually overlap (and their difference is due to sequencing errors) or they originate from repeat regions (repeats make layout problem very difficult). Mate pair reads help to alleviate this problem.

3) Consensus

Derive the final sequence from the layout by correcting errors in sequence reads (most frequent nucleotide in the layout given enough sequencing reads cover the region).

Assembly with Velvet and Abyss, QC

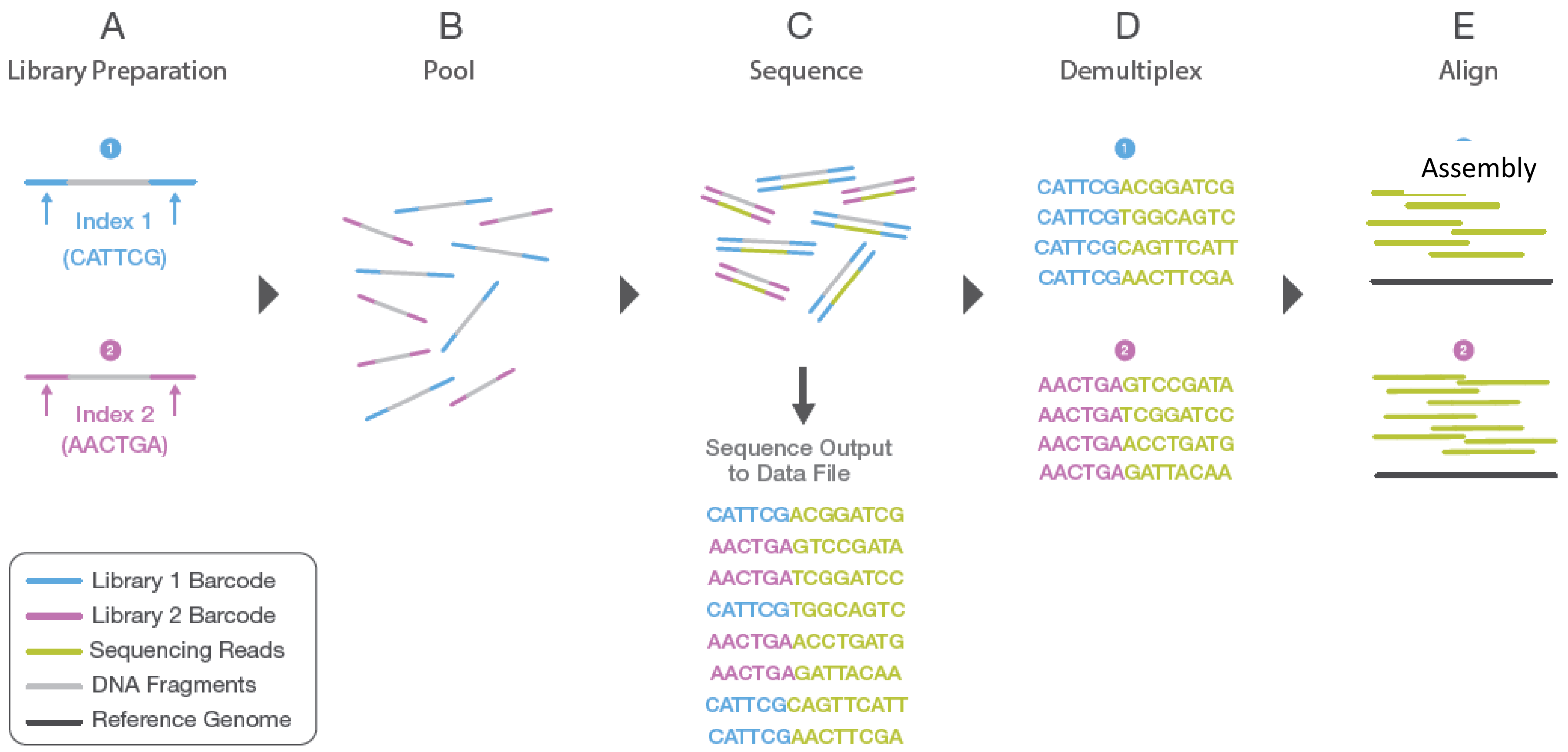
Very simple assembly of some reads from a mitochondrial genome.

```
#  Prepare the data. Unzip the file.
$ cd assembly_taster
$ mkdir results
```

Use fastQC for QC statistics and a quick graphical overview of the dataset.

```
# Run FastQC on the dataset
$ fastqc -o results mt_reads.fastq

# open report.html in browser by double clicking
```

Barcoding/Multiplexing



Adapted from Illumina.com

Multiplexing allows that more samples to be pooled and sequenced simultaneously during a single sequencing run.
This techniques reduces the cost and the time of analysis.

Demultiplexing and Trimming

De-multiplexing based on the adaptor sequence:

```
# fastx splitter splits mt_reads.fastq by barcode.
# --bol indicates that the barcodes are at the 5’ end.
# Note the following command should be typed on a single line:

$ fastx_barcode_splitter.pl < mt_reads.fastq --bcfile
mt_barcodes.txt --bol --suffix .fastq --prefix results/
```

Trimming removes the barcodes sequences corresponding to the multiplexing adaptor from the beginning or end of the read.

```
$ cd results
$ fastx_trimmer -i mt1.fastq -f 8 -o trimmed_mt1.fastq -Q33
```

Quality Filtering

Removing low quality sequences increases the accuracy of the assembly.

```
$ fastq_quality_filter -i trimmed_mt1.fastq -q 25 -p 80 -o
qual_trim_mt1.fastq -Q33 -v
```

Removing artifacts from the sequencing and low quality reads will contribute to a better assembly quality.

Assembly with Velvet

Velvet is a highly popular short read assembler.
‘k’ == Kmer length i.e. the length of sub sequences that the data is being broken up into, and is often one of the most important parameters to manipulate.

```
$ velveth velvet_k21 21 -short -fastq qual_trim_mt1.fastq
$ velvetg velvet_k21 -read_trkg -yes amos_file yes

# inspect the results with the program tablet
$ tablet velvet_k21/velvet_asm.afg &
```

Assembly is an iterative process of testing multiple parameters values.
VelvetOptimiser is a script which automatically tries multiple parameter combinations and returns the best assembly it can find.

```
$ velvetoptimiser -s 27 -e 31 -f '-short -fastq
qual_trim_mt1.fastq' -a 1
$ tablet auto_data_31/velvet_asm.afg &
```

Assembly With Abyss

```
# Running abyss in single end mode with k=21
$ abyss -k21 qual_trim_mt1.fastq -o abyss_contigs.fa

# try abyss with multiple kmer values
$ for k in {15..20}
do
    abyss -k$k qual_trim_mt1.fastq -o abyss_k$k.fa
done
# create an abyss directory and copy the resulting files
```

Assembly With Abyss

```
# Running abyss in single end mode with k=21
$ abyss -k21 qual_trim_mt1.fastq -o abyss_contigs.fa

# try abyss with multiple kmer values
$ for k in {15..20}
do
    abyss -k$k qual_trim_mt1.fastq -o abyss_k$k.fa
done
# create an abyss directory and copy the resulting files
```