

# BIO392 file formats and tools

Izaskun Mallona

- Commands/options are in typewriter font
- URLs are highlighted in blue

- Lecture from SIB

[https://edu.sib.swiss/pluginfile.php/2878/mod\\_resource/content/4/couselab-html/content.html](https://edu.sib.swiss/pluginfile.php/2878/mod_resource/content/4/couselab-html/content.html)

- Run the exercises till number 4

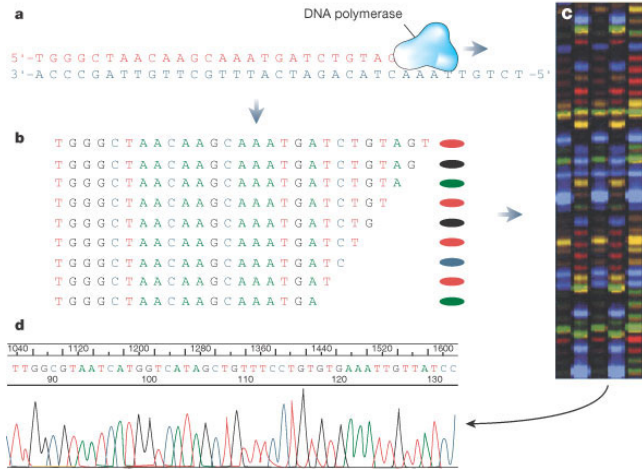
# Commonly used formats

- Reference genomes
- Fasta and FastQ (Unaligned sequences)
- SAM/BAM (Alignments)
- BED (Genomic ranges)
- GFF/GTF (Gene annotation)
- BEDgraphs (Genomic ranges)
- Wiggle files, BEDgraphs and BigWigs (Genomic scores).
- Indexed BEDgraphs/Wiggles
- VCFs (variants)

# Reference genomes

- Reference genomes describe the 'consensus' DNA sequence
- (Who is the human consensus for DNA sequencing?)
- Aside of human variation, multiple assemblies have been released

# Sanger sequencing Nature 409, 863 (2001)







GRCh stands for 'Genome Reference Consortium'

- Human GRCh37 (hg19)
- Human GRCh38
- Mouse mm10
- Mouse GRCm38
- Zebrafish, chicken and others:  
<https://www.ncbi.nlm.nih.gov/grc> The Genome Reference consortium

# Reference genomes: FASTA

- A reference genome is a collection of contigs/scaffolds
- A contig is a stretch of DNA sequence encoded as A,G,C,T,N.
- Typically comes in FASTA format.
- ">" line contains the scaffold name
- Following lines contain the sequence (single line, 80 nt-column sized...)

# Reference genomes: FASTA

```
>NC_009902.1 Babesia bovis T2Bo mitochondrion, complete genome
TTTAAAAAAGTGTTAAAAA CTTTATACATTAAAAAATTTAAACAAAGTGATCATGTATAAAGTACACTTGT
TACTGTGTAATATCAAAAACAATTTAATTTCAAAAATTTTGAAATATGTTTTTGTGTTGTGTTATAAA
GTTTTTTTTCAAAATTATATATGTTTGCATTTGCTGGATATAGTTCGGTCTCTGCAAA CCATAAAGTCAT
CGGTATATCCTACATATGGCTTTCATATTGTTTGGAGTTATTGGATTTTATATGAGTATTTTGATAAGA
ACAGAATTGAGTATGAGTGTTTAAAGATTATGACAATGGATACTCTTGAGATATACAATATGATGTTTT
```

# Patches, alternate loci and primary assembly

- Primary assembly: the best known assembly of a haploid genome
  - Chromosome assembly
  - Unlocalized sequence (associated to a chromosome but whose order/orientation is unknown)
  - Unplaced sequence (not linked to any chromosome)
- Alternate loci: An alternate representation of a locus (usually highly polymorphic regions, such as the MHC region)
- Patches: A contig sequence that is released outside of the full assembly release
  - Fix: error correction
  - Novel: new sequences that will be included into the next full assembly release

# Browsing genomic patches

- Activity: browse genomic patches, i.e. [ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/000/001/405/GCA\\_000001405.27\\_GRCh38.p12/README\\_patch\\_release.txt](ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/000/001/405/GCA_000001405.27_GRCh38.p12/README_patch_release.txt)

# Retrieving fasta sequences manually (UCSC)

- Try to retrieve the DIEXF gene promoter
- (What is a promoter in terms of sequence?)
- Go to an assembly <https://genome-euro.ucsc.edu/cgi-bin/hgTracks?db=hg38>
- Query gene symbol (i.e. DIEXF)
- Click into the gene (gencode track)
- Click into the sequence and links item
- Specify your promoter definition

# Manually downloading the DEXF promoter

← → ↺ [https://genome-euro.ucsc.edu/cgi-bin/hgGene?hgg\\_gene=uc001hhr.3&hgç](https://genome-euro.ucsc.edu/cgi-bin/hgGene?hgg_gene=uc001hhr.3&hgç) ☆

Genomes Genome Browser Tools Mirrors Downloads My Data He

## Human Gene DEXF (ENST00000491415.6) Description and Page Index

**Description:** Homo sapiens digestive organ expansion factor homolog (zebrafish) (DEXF), mRNA  
**Gencode Transcript:** ENST00000491415.6  
**Gencode Gene:** ENSG00000117597.17  
**Transcript (Including UTRs)**  
**Position:** hg38 chr1:209,828,007-209,857,565 **Size:** 29,559 **Total Exon Count:** 12 **Strand:** +  
**Coding Region**  
**Position:** hg38 chr1:209,828,064-209,851,447 **Size:** 23,384 **Coding Exon Count:** 12


<b>Page Index</b>	Sequence and Links	UniProtKB Comments	CTD	RNA-Seq Express
RNA Structure	Protein Structure	Other Species	GO Annotations	mRNA Description
Other Names	Methods			

**Data last updated:** 2016-03-28

### Sequence and Links to Tools and Databases

Genomic Sequence (chr1:209,828,007-209,857,565)			mRNA (may differ from genome)		Protein
Gene Sorter	Genome Browser	Other Species FASTA	Gene interactions	Table Schema	BioGP
CGAP	Ensembl	Entrez Gene	ExonPrimer	GeneCards	Gepis
HGNC	HPRD	Lynx	MGI	MOPED	neXtP
PubMed	Reactome	Stanford SOURCE	UniProtKB		

# Manually downloading the DIEXF promoter

 Genomes Genome Browser Tools Mirrors Downloads My Data

## Genomic Sequence Near Gene

### Get Genomic Sequence Near Gene

Note: if you would prefer to get DNA for more than one feature of this track at a time, try the [Table Browser](#) using the output format sequence.

#### Sequence Retrieval Region Options:

- ☒ Promoter/Upstream by  bases
- ☐ 5' UTR Exons
- ☐ CDS Exons
- ☐ 3' UTR Exons
- ☐ Introns
- ☐ Downstream by  bases
- ☒ One FASTA record per gene.
- ☐ One FASTA record per region (exon, intron, etc.) with  extra bases upstream (5') and  extra downstream (3')
- ☐ Split UTR and CDS parts of an exon into separate FASTA records

Note: if a feature is close to the beginning or end of a chromosome and upstream/downstream bases are added, they may be truncated in order to avoid extending past the edge of the chromosome.



# Manually downloading the DLEXF promoter

```
>hg38_knownGene_uc001hhr.3 range=chr1:209827007-209827008
gtttctgctgtttgttaaattggggaatgctggaacagatttgtttgcgggg
actcttccaatactttcagaaaatgcgagaataggggtgaggggtgggaatc
tcagacttgtagggcccatgattgatataaacacacacaggcggcagacc
taatgggtaaaagcatgtgttgcatcagttaaggtttttctctcttctc
ttgctagcgtgttatcttttcttttcttttcttttcttttttttttttctg
agatggagtcctagcttttgcgcccaggctggagtaggctggagtgagtg
ggagtgatctcggtcattgcaacctccacctcccggttccagcgattc
tcctgcctcacctcctgagtagctgggattacaggcgcccgcctaccacgc
ccggctgatttttgtacttttagtagagacgggggtttcaccatgtttggc
catgctggtctcgaactcctgacctcaggtgatccgcccatctcggcctc
ccaaagtgttgagattacaggcgtagccaccgcgcccggccgctagcgt
gttatcttttctaagcatcagtttccttatctgcaacaccaggcttatta
acaagacctatctgtacactgtttgtggtgatgaagtgagatgttcaggca
cccttaaatgtttggttgatattttattgcagtatactgtaaagtcactg
cattcgactatctccgctactacacatttacgcagactgatttccataac
caaaacacaagcacaaagctcatgccccgactcacgcaaccgggaagc
ccagctgcccacgttctagggctctgagaacactagtgaacgaactcccg
tgctttcaaagagctgcggtagggggcagaaccgggaaccggatgttcta
agcctgtcgtacgagcgcgacgtaaagcggtatctgctttatggcaccttg
ctttcgccgtaaagcgcagtcagcgagccacgtgcttggttgactgga
```

# How do we do this in a reproducible manner?

- Scripting. We store an up-to-date reference genome in our computer (once)...
- ... and then use specific file standards to specify the genome annotation (i.e. GTF, BED files).
- Activity: read the documentation of UCSC on how to download sequences  
<http://genome.ucsc.edu/FAQ/FAQdownloads.html> (section Extracting sequence in batch from an assembly)

# How to specify the DIEXF promoter in a reproducible manner?

Same concept that when we did manually:

- 1 Download the human genome sequence
- 2 Download a file with all the genes (transcripts) locations (not sequences, but their coordinates)
- 3 Then select the gene we are looking for (DIEXF)
- 4 Decide what a promoter is (i.e. 2 kb upstream of the gene) and update the coordinates accordingly
- 5 Then use a specific tool to slice the full genome to only report the DIEXF promoter

# How to specify the DIEXF promoter? standards

Same concept that when we did manually + some standards

- 1 Download the human genome sequence (fasta)
- 2 Download a file with all the genes (transcripts) locations (GTF)
- 3 Then select the gene we are looking for (DIEXF) (grep/awk)
- 4 Decide what a promoter is (i.e. 2 kb upstream of the gene) and update the coordinates accordingly (BEDfile, bedtools/awk)
- 5 Then use a specific tool to slice the full genome to only report the DIEXF promoter (bedtools)

- This we can do because the genome consortia and the science community release free data, software/toolsets
- We benefit from the Unix-like operating systems
- But the access to the same data and tools is not enough: the need for data standards

# Commonly used formats

- **Fasta and FastQ**
- SAM/BAM (Alignments)
- BED (Genomic ranges)
- GFF/GTF (Gene annotation)
- BEDgraphs (Genomic ranges)
- Wiggle files, BEDgraphs and BigWigs (Genomic scores).
- Indexed BEDgraphs/Wiggles
- VCFs (variants)

# Short reads sequencing

- Sequencing very short reads (50 to 150 nucleotides) is common practice
- We get hundreds of millions of short reads for each experiment
- Instead of assembling them, we map them into a reference genome
- Activity: read <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2836519/>
- Sequencers provide sequence and error rates assessment: fasta format is not suitable, but fastq is

# FASTQ: Short read sequencing

- Next step to FASTA: including quality data
- Standard de facto for short read, high-throughput sequencing instruments such (i.e. Illumina)

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345 length=36
GGGTGATGGCCGCTGCCGATGGCGTCAAATCCCACC
+
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII9IG9IC
```

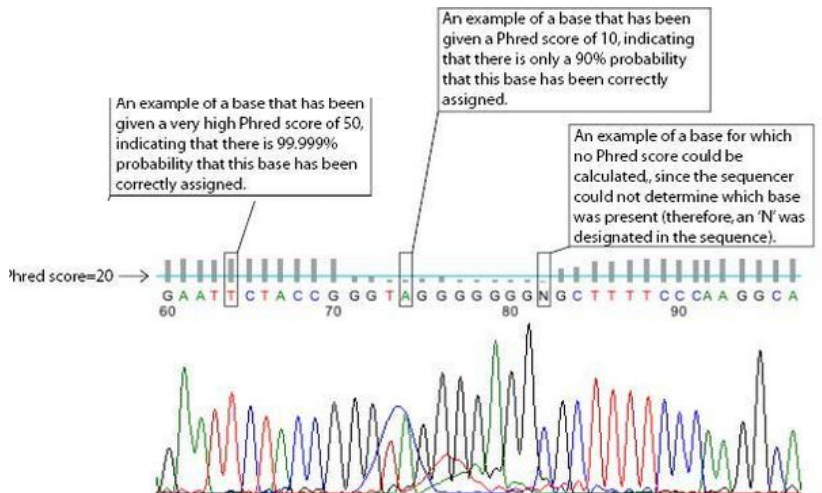


- Sequence quality is represented using Phred scores
- The sequencing quality score of a given base  $Q$  is defined by as
- $Q = -10 \log_{10} P$

**Phred quality scores are logarithmically linked to error probabilities**

<b>Phred Quality Score</b>	<b>Probability of incorrect base call</b>	<b>Base call accuracy</b>
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10,000	99.99%
50	1 in 100,000	99.999%
60	1 in 1,000,000	99.9999%

# phred scores (old school Sanger electrophoretogram)



# Phred scores encoding

- There are several Phred score encodings:
- Activity: read about the Quality scores offsets at [https://en.wikipedia.org/wiki/FASTQ\\_format](https://en.wikipedia.org/wiki/FASTQ_format) and [https://wiki.bits.vib.be/index.php/Identify\\_the\\_Phred\\_scale\\_of\\_quality\\_scores\\_used\\_in\\_fastQ](https://wiki.bits.vib.be/index.php/Identify_the_Phred_scale_of_quality_scores_used_in_fastQ)

# Phred scores encoding (Wikipedia)

```
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....
.....XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
.....IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
.....JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
|
33          59   64   73          104          126
0.....26...31.....40
          -5...0.....9.....40
          0.....9.....40
          3.....9.....41
0.2.....26...31.....41
```

- S - Sanger            Phred+33, raw reads typically (0, 40)
- X - Solexa           Solexa+64, raw reads typically (-5, 40)
- I - Illumina 1.3+ Phred+64, raw reads typically (0, 40)
- J - Illumina 1.5+ Phred+64, raw reads typically (3, 41)  
with 0=unused, 1=unused, 2=Read Segment Quality Control Indicator (bold)  
(Note: See discussion above).
- L - Illumina 1.8+ Phred+33, raw reads typically (0, 41)

# Unaligned sequences (from sequencers): FASTQ

- FASTQs stands for FASTA with Qualities
- Plain text files with chunks of four lines:
  - @ identifier line
  - Sequence
  - "+" (sometimes the sequence name, again)
  - Quality scores (different encodings exist)

# Example FASTQ entry

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345 length=36
GGGTGATGGCCGCTGCCGATGGCGTCAAATCCCACC
+
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII9IG9IC
```

- Activity: FASTQ/A exercises (exercises 5 to 14)



# Working with fastq files

```
## retrieving an example fastq file
curl https://molb7621.github.io/workshop/_downloads/SP1.fq \
  > file.fastq

## counting number of reads
awk 'END{print NR/4}' file.fastq

## transforming into fasta
awk 'NR % 4 == 1 {print ">"$1};
     NR % 4 == 2 {print}' file.fasta
```

# Further manuals on awk

- <https://en.wikipedia.org/wiki/AWK>
- <http://www.grymoire.com/Unix/Awk.html>

# awk: Counting the number of items in a fastq

So in fastq each data chunk is stored in four different lines. We'll need to be able to extract the first, second, third or fourth line for each block of four lines. Using awk,

```
awk 'END{print NR/4}' file.fastq
```

- NR gives the number of records (line numbers)
- FASTQ are chunks of 4 lines for each sequence
- NR/4 at the END of the file indicates the number of sequences

# Working with fastq files

```
## retrieving an example fasta file
curl https://molb7621.github.io/workshop/_downloads/SP1.fq \
  > file.fastq

## counting number of reads
awk 'END{print NR/4}' file.fastq

## transforming into fasta
awk 'NR%4==1{a=substr($0,2);} NR%4==2 {print ">"a"\n"$0}' \
  file.fastq
```

```
awk 'NR%4==1{a=substr($0,2);} NR%4==2 {print ">"a"\n"$0}' \
file.fastq
```

- % is a modulo operator
- `NR%4==1` will retrieve the first line of a fastq chunk (header)
- `NR%4==2` will retrieve the second line (the sequence)
- the id line will be prepended with the `>` and reduced to a substring (chopped)
- This will be applied to all lines!

# Still need to align the FASTQ reads to the reference genome

- Discussion: how to get rid of the sequences and to have a smaller data representation?
- Trying to transform sequence to reference genome coordinates (= aligning to the genome/mapping)
- i.e. transforming ACGCACGCACGCACGCCCC to human genome hg19 'chr10:10010-10030'

- SAM - Sequence Alignment Map.
- The standard stores where the reads (i.e. the ones we had as FASTQs) map in the reference genome
- Recognised by majority of software and browsers: standard

# What is an alignment?

- Sequence alignment: arrange a set of sequences to identify regions of similarity/identity
- Mapping short reads against a reference genome: aligning large amounts short reads to a reference genome



# Local alignments vs global alignment

1	2	3	4	5	6	7	8	9	10	11
C	G	T	C	C	G	A	A	G	T	G
			.							
★	★	T	A	C	G	A	A	★	★	★

(a) Global alignment

3	4	5	6	7	8
T	C	C	G	A	A
	.				
T	A	C	G	A	A

(b) Local alignment

1	2	3	4	5	6	7	8
C	G	T	C	C	G	A	A
			.				
★	★	T	A	C	G	A	A

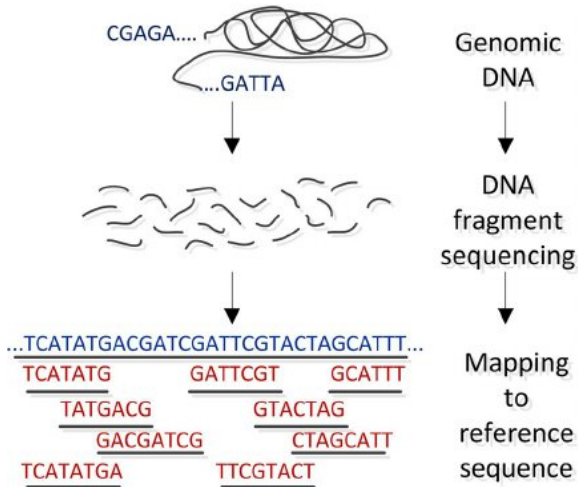
(c) Semi-global alignment

Alachiotis et al, 2013

- Chromosome
- Locus (coordinate)
- CIGAR string, i.e.
- 30M1D2M - 30 bases continuously match, 1 deletion from reference, 2 base match
- Some flags (<https://broadinstitute.github.io/picard/explain-flags.html>)

- Activity: read a post on CIGAR encoding <http://bioinformatics.cvr.ac.uk/blog/tag/cigar-string/>  
(please skip the Java code)

# Next generation sequencing to SAM



Pavlopoulos et al 2013

# Next generation sequencing to SAM

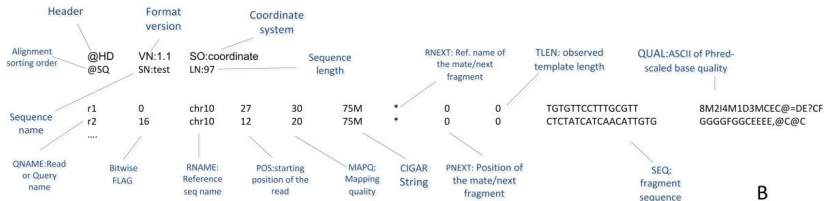
starting symbol      sequence identifier  
@HWI-EAS3X\_10102\_2\_120\_19829\_1823#0/2  
sequence end      sequence  
start QS      TCTAACTTACTTAGCATAGCTGTTAAAATTTTGAGTT  
+ (optionally the same identifier)  
DEAEE:B:BE5EEEEED=:DEA:-AE5DDBDFFEDEEDFAE      quality score

Pavlopoulos et al 2013

# Next generation sequencing to SAM

Coordinates 123456789...  
Reference AAATGAATAATCTCTATCATCAACATTGTGTTCTTGCCTTTTAACCTTTCCT  
Reads r1 CTCTATCATCAACATTGTG  
r2 TGTGTTCTTGCCTTT

A



B

Pavlopoulos et al 2013

- Activity: read the SAM format specification
- <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2723002/>

- Exercise number 15



# From SAM to BED: counts

- BED files are simpler data representations, usually the next step after getting the SAM files
- Why? they are smaller and easier to handle
- For instance, after mapping a new genome-wide sequencing BED files with the genomic coverages are generated
- Discussion: how to handle expression data, i.e. transcripts without introns etc? how do we count them?
- Activity: read <https://bedtools.readthedocs.io/en/latest/content/tools/genomecov.html>

# Keep it simple: count and transform into BED files

- BED (Browser Extensible Data) files come in different flavours
- BED3: 3 tab separated columns, chromosome (scaffold), start, end
- BED6: BED3 plus name, score, strand

```
chr22 1000 5000  
chr22 2000 6000
```

```
chr22 1000 5000 cloneA 960 +  
chr22 2000 6000 cloneB 900 -
```

```
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
```

# How do we count? 0s and 1s

- Even though BED files are standard how to count nucleotides is not
- 0-start vs. 1-start : Does counting start at 0 or 1?
- For a counted range, is the specified interval fully-open, fully-closed, or a hybrid-interval (e.g., half-open)?
- Activity: read <http://genome.ucsc.edu/blog/the-ucsc-genome-browser-coordinate-counting-systems/>

- Exercises 16 to 18

# The need for further data formats

- So to sum up until now generally we have a reference genome, reads that were retrieved as FASTQ files, mapped and transformed to SAM files
- (Which step was the last one with actual sequences on them?)
- So, at last, we can answer questions
- Which fraction of the human genome is covered by exons?
- Genomic locations of SNPs associated with prostate cancer?
- Are gene bodies more variable (in terms of SNPs) than intergenic regions?
- For this we need further data analysis tools (i.e. BEDtools) on BED files and other data formats (annotations and variation)



# What does annotation mean?

- Genomic annotations are layers to genomic coordinates specifying their nature



Henrik Lantz, BILS/SciLifeLab

# How to store genomic annotations? GFF3

<u>Segid</u>	source	type	start	end	score	strand	phase	attributes
Chr1	Snap	gene	234	3657	.	+	.	ID=gene1; Name=Snap1;
Chr1	Snap	mRNA	234	3657	.	+	.	ID=gene1.m1; Parent=gene1;
Chr1	Snap	exon	234	1543	.	+	.	ID=gene1.m1.exon1; Parent=gene1.m1;
Chr1	Snap	CDS	577	1543	.	+	0	ID=gene1.m1.CDS; Parent=gene1.m1;
Chr1	Snap	exon	1822	2674	.	+	.	ID=gene1.m1.exon2; Parent=gene1.m1;
Chr1	Snap	CDS	1822	2674	.	+	2	ID=gene1.m1.CDS; Parent=gene1.m1;
		start_codon						Alias, note, ontology_term ...
		stop_codon						

Henrik Lantz, BILS/SciLifeLab

# How to store genomic annotations? GTF

<u>Seqid</u>	source	type	start	end	score	strand	phase	attributes
Chr1	Snap	exon	234	1543	.	+	.	gene_id "gene1"; transcript_id "transcript1";
Chr1	Snap	CDS	577	1543	.	+	0	gene_id "gene1"; transcript_id "transcript1";
Chr1	Snap	exon	1822	2674	.	+	.	gene_id "gene1"; transcript_id "transcript1";
Chr1	Snap	CDS	1822	2674	.	+	2	gene_id "gene1"; transcript_id "transcript1";
		start_ codon						
		stop_ odon						

Henrik Lantz, BILS/SciLifeLab

# Why so complex? Open reading frames

```

      N  V  P  V  N  I  *  I  I  V  M  P  K  V  E
      K  C  P  C  *  N  *  H  N  S  M  V  *  G
      *  L  P  M  L  E  L  S  Q  E  H  S  L  *
=====
3'-LVVLGLCCGLVLVGLLVCLVVGGLVCGGVLCGGV-5'
5'-ATTTACAGGGCATTAATCTAATGATTGCTCATGGCTTAGCCT-3'
=====
      I  Y  *  G  I  N  S  N  D  C  S  W  L  S  L
      F  T  G  A  L  I  L  M  I  A  H  G  L  A
      L  Q  G  H  *  F  *  W  L  L  M  A  *  P
```

Steven M. Carr

# Why so complex? CDS, exons, introns, stop codons

- How many transcript does a gene have?
- Are they tissue dependent?
- How can we annotate the different between transcription and translation?

- Reading: <https://www.ensembl.org/info/website/upload/gff.html>

chromA	chromStartA	chromEndA	dataValueA
chromB	chromStartB	chromEndB	dataValueB
chr19	49303800	49304100	0.50
chr19	49304100	49304400	0.75
chr19	49304400	49304700	1.00

- To display continuous-valued data in track format.
- Useful for probability scores

# Which are the differences between BEDgraphs and BED?

- BED, BED, BED12?
- Advantages: the coordinates are specified, so sparsity is allowed
- Next step in file formats: trying to cover all the genome (that is, no sparsity anymore)
- Example: does it make sense to generate a BED file with GC content? ([GC content at Wikipedia](#))
- How can we store features with definite start and ends but for which the value is the primary purpose, but not their starts and ends?



- To display continuous-value data
- GC percent, probability scores, and transcriptome data.
- Data is not sparse! Wiggle data elements must be equally sized (step)

```
variableStep chrom=chr2  
300701 12.5  
300702 12.5  
300703 12.5  
300704 12.5  
300705 12.5
```

# Wig with added span

```
variableStep chrom=chr2 span=5  
300701 12.5
```

# Wig with fixedStep and span

```
fixedStep chrom=chr3 start=400601 step=100 span=5  
11  
22  
33
```

- Read more on Wig files at <https://genome.ucsc.edu/goldenpath/help/wiggle.html>

# On coordinates, 0s vs 1s and open and closed intervals

- Remember the unusual 0 and 1 counting when converting BED, Wiggle and/or when using genomic positions at the UCSC genome browser (very last part of the article <http://genome.ucsc.edu/blog/the-ucsc-genome-browser-coordinate-counting-systems/>)

- Run exercises 16 to 24

- Standard file format for storing variation data
- Unambiguous, scalable and flexible
- 8 columns:
  - 1 CHROM
  - 2 POS
  - 3 ID
  - 4 REF
  - 5 ALT
  - 6 QUAL
  - 7 FILTER
  - 8 INFO



#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	NA19909
11	5248232	rs334	T	A	100	PASS	AA=T   ;AC=1;AF=0.0273562;AFR_AF=0.0998;AMR_AF=0.0072;AN=2;DP=22876;EAS_AF=0;EUR_AF=0;EX_TARGET;NS=2504;SAS_AF=0;VT=SNP	GT	0 1

EMBL/EBI training

# Quality values: which one?

- Phred-scaled quality score for the assertion made in ALT. i.e.  
 $Q = -10 \log_{10} P$  being  $P(\text{call in ALT is wrong})$
- Read quality
- Mapping quality
- Variant calling quality

- Lecture by Michael Lawrence (VariantExplore package)
- [https://www.bioconductor.org/help/course-materials/2014/CSAMA2014/3\\_Wednesday/lectures/VariantCallingLecture.pdf](https://www.bioconductor.org/help/course-materials/2014/CSAMA2014/3_Wednesday/lectures/VariantCallingLecture.pdf)

# The VCF format

- Activity: read <http://www.internationalgenome.org/wiki/Analysis/Variant%20Call%20Format/vcf-variant-call-format-version-40>

- Run exercises from 25 on