Studying the transcriptome using RNA-seq

Cecilia Coimbra Klein

IBUB
Institut de Biomedicina
de la Universitat de Barcelona

UNIVERSITAT DE BARCELONA

CRG
Centre
for Genomic
Regulation

UVIC UNIVERSITAT DE VIC
UNIVERSITAT CENTRAL DE CATALUNYA

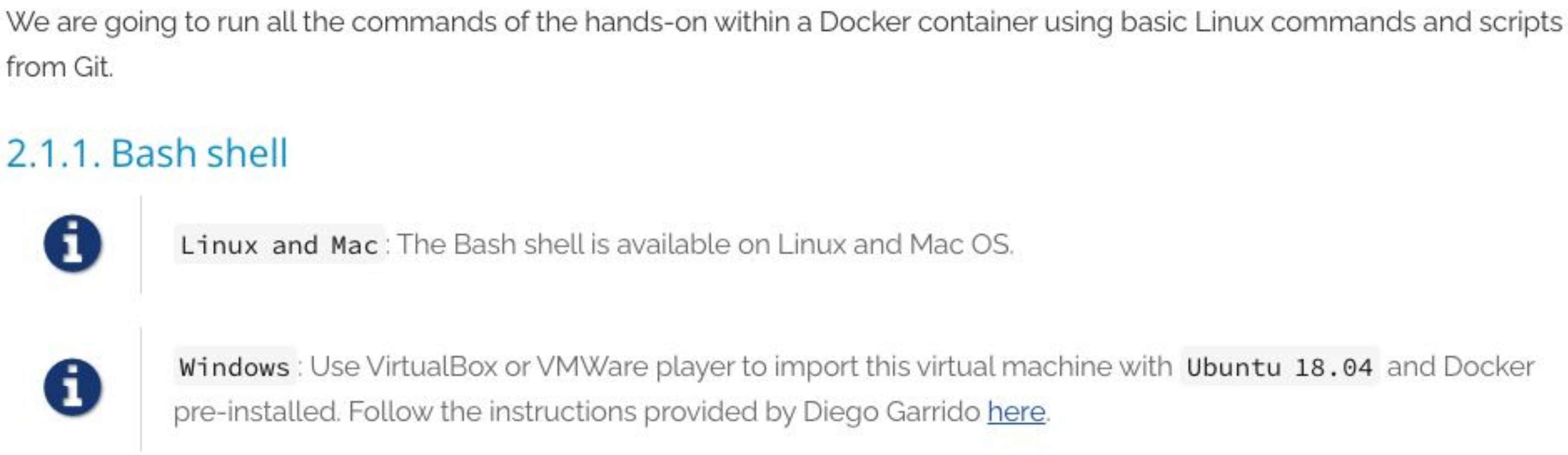
Master in Omics
Data Analysis

Outline

1. Introduction
2. Basic concepts
   2.1. Hands-on:
      2.1.1. Basic Linux Commands
      2.1.2. Git and GitHub
      2.1.3. Docker
   2.2. RNA-seq:
      2.2.1. RNA biology
      2.2.2. NGS technologies
      2.2.3. RNA-seq experimental design
      2.2.4. Reference gene annotation
      2.2.5. Data formats
3. Short-read RNA-seq data processing
4. Gene level RNA-seq data analysis
5. Isoform level RNA-seq analyses
6. Regulation of gene expression

Basic Linux commands

Basic Linux commands



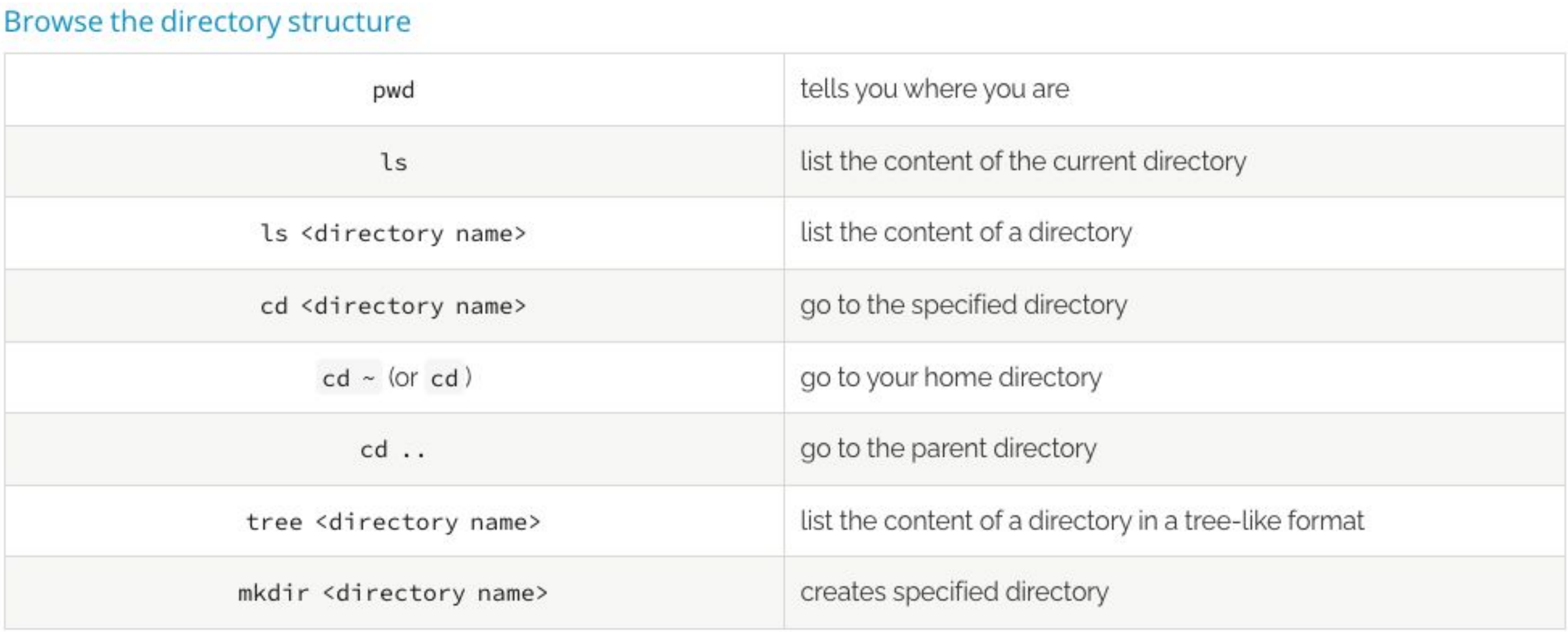
We are going to run all the commands of the hands-on within a Docker container using basic Linux commands and scripts from Git.

2.1.1. Bash shell

`Linux and Mac` : The Bash shell is available on Linux and Mac OS.

`Windows` : Use VirtualBox or VMWare player to import this virtual machine with `Ubuntu 18.04` and Docker pre-installed. Follow the instructions provided by Diego Garrido here.

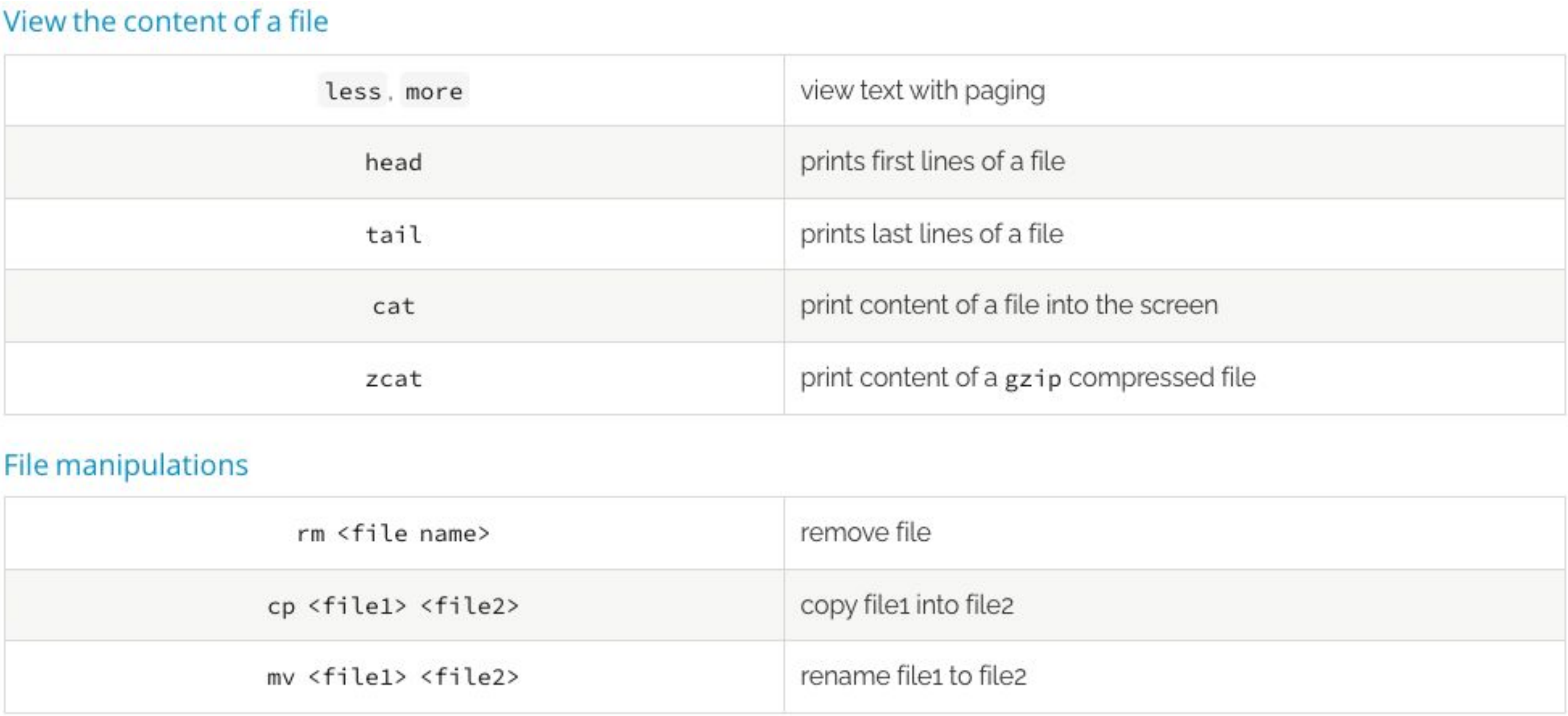
Basic Linux commands



Browse the directory structure

`pwd`	tells you where you are
`ls`	list the content of the current directory
`ls <directory name>`	list the content of a directory
`cd <directory name>`	go to the specified directory
`cd ~` (or `cd`)	go to your home directory
`cd ..`	go to the parent directory
`tree <directory name>`	list the content of a directory in a tree-like format
`mkdir <directory name>`	creates specified directory

Basic Linux commands



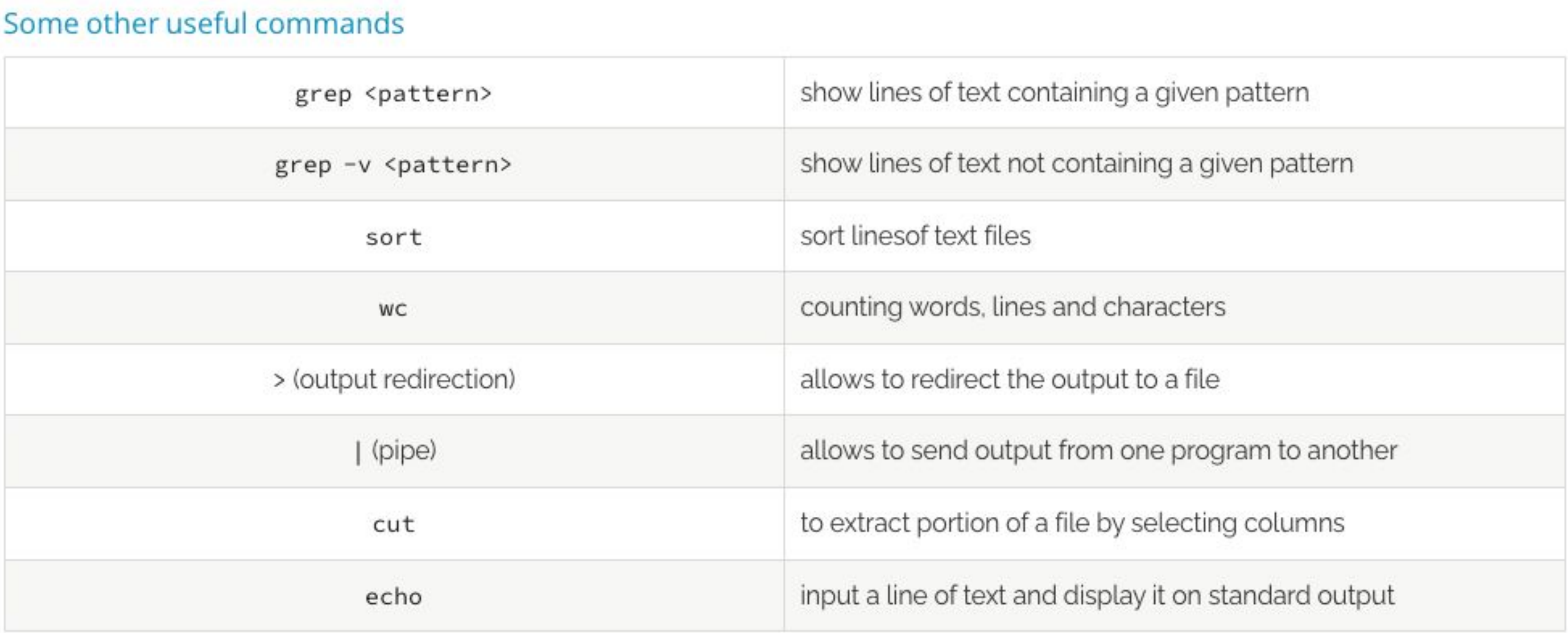
View the content of a file

`less`, `more`	view text with paging
`head`	prints first lines of a file
`tail`	prints last lines of a file
`cat`	print content of a file into the screen
`zcat`	print content of a `gzip` compressed file

File manipulations

`rm <file name>`	remove file
`cp <file1> <file2>`	copy file1 into file2
`mv <file1> <file2>`	rename file1 to file2

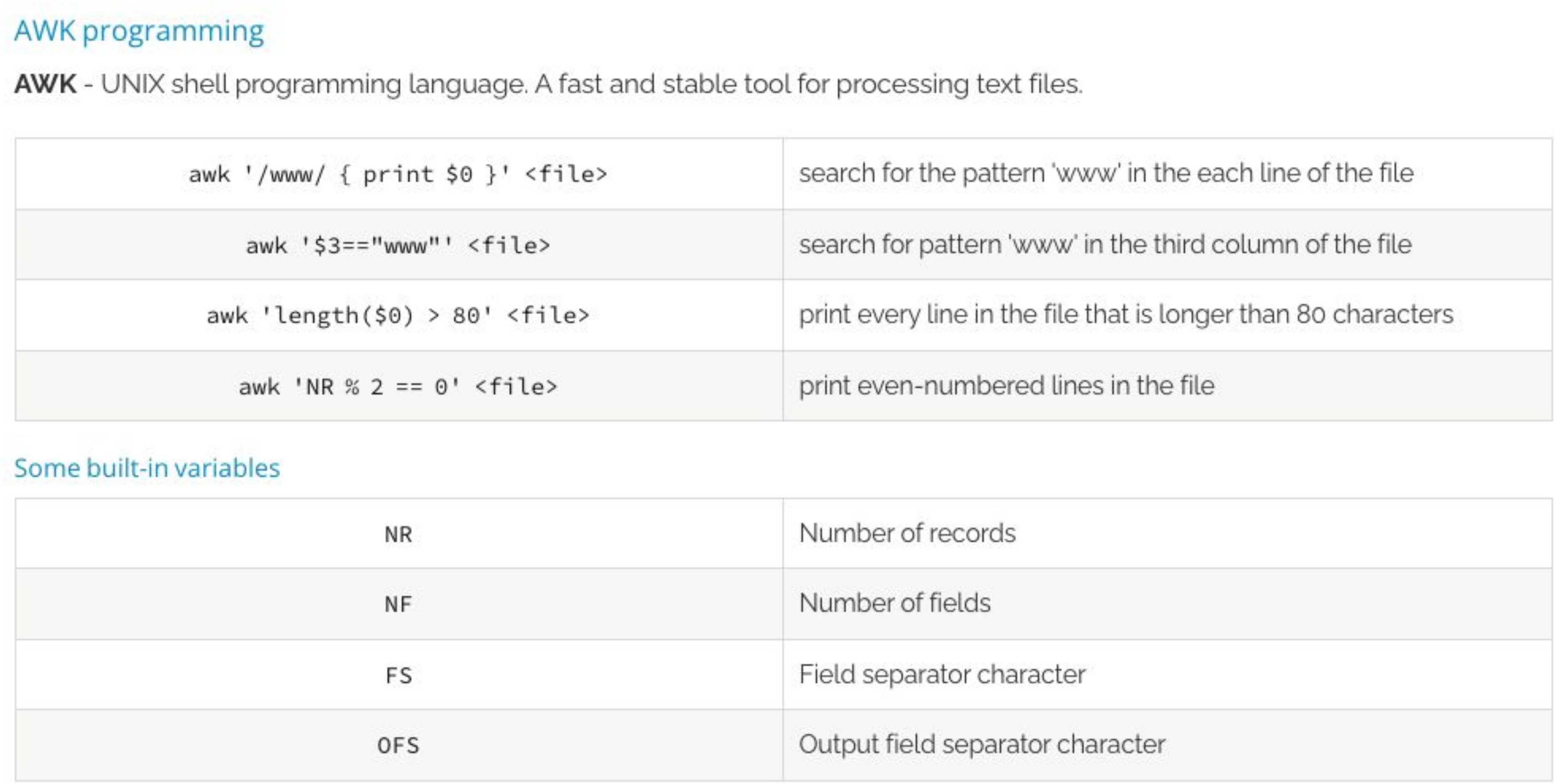
Basic Linux commands



Some other useful commands

`grep <pattern>`	show lines of text containing a given pattern
`grep -v <pattern>`	show lines of text not containing a given pattern
`sort`	sort linesof text files
`wc`	counting words, lines and characters
> (output redirection)	allows to redirect the output to a file
| (pipe)	allows to send output from one program to another
`cut`	to extract portion of a file by selecting columns
`echo`	input a line of text and display it on standard output

AWK programming



AWK programming

AWK - UNIX shell programming language. A fast and stable tool for processing text files.

`awk '/www/ { print $0 }' <file>`	search for the pattern 'www' in the each line of the file
`awk '$3=="www"' <file>`	search for pattern 'www' in the third column of the file
`awk 'length($0) > 80' <file>`	print every line in the file that is longer than 80 characters
`awk 'NR % 2 == 0' <file>`	print even-numbered lines in the file

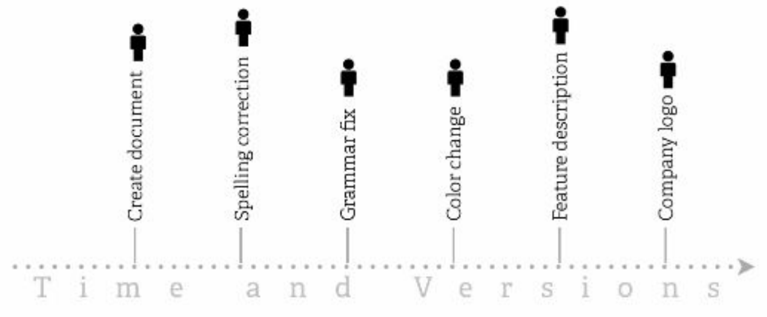
Some built-in variables

`NR`	Number of records
`NF`	Number of fields
`FS`	Field separator character
`OFS`	Output field separator character

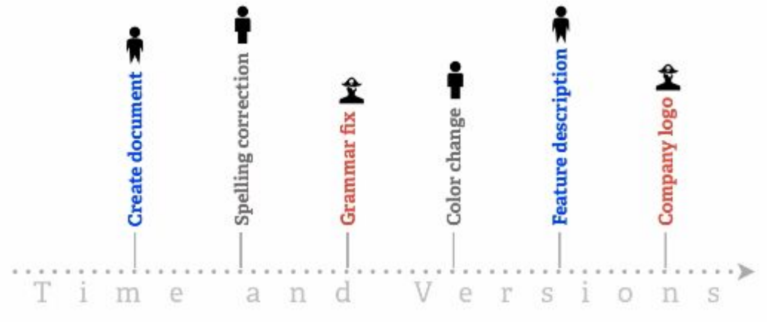
Basics Git and GitHub

Basics Git and GitHub

- **Git** is a *fast* and *modern* implementation of **version control**.
- **Git** provides **history** of content change.



- **Git** facilitates **collaborative changes** to files.

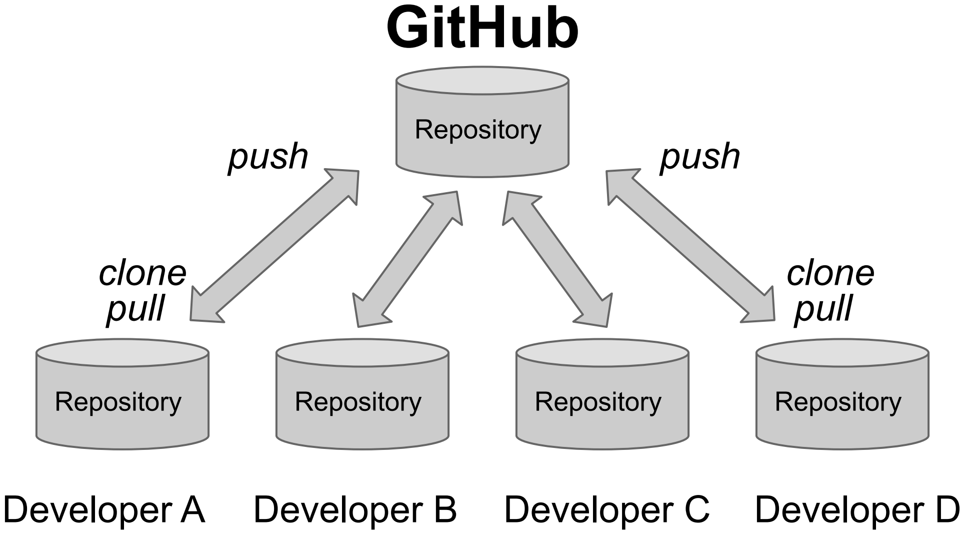


https://git-scm.com/video/what-is-version-control

Basics Git and GitHub

Git is the free and open source distributed **version control** system that's responsible for everything **GitHub** related that happens locally on your computer.

GitHub is the most widely used web-based hosting service for **version control** using **Git**.



Basics Docker

Basics Docker

Reproducibility

- **Docker** provides the ability to package and run an application in a loosely isolated environment called a **container**.
- **Containers** are lightweight and **contain everything needed** to run the application, so you do not need to rely on what is currently installed on the host.
- You can easily **share containers** while you work, and be sure that everyone you share with gets the **same container that works in the same way**.

Basics Docker

IMAGES

Docker images are a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

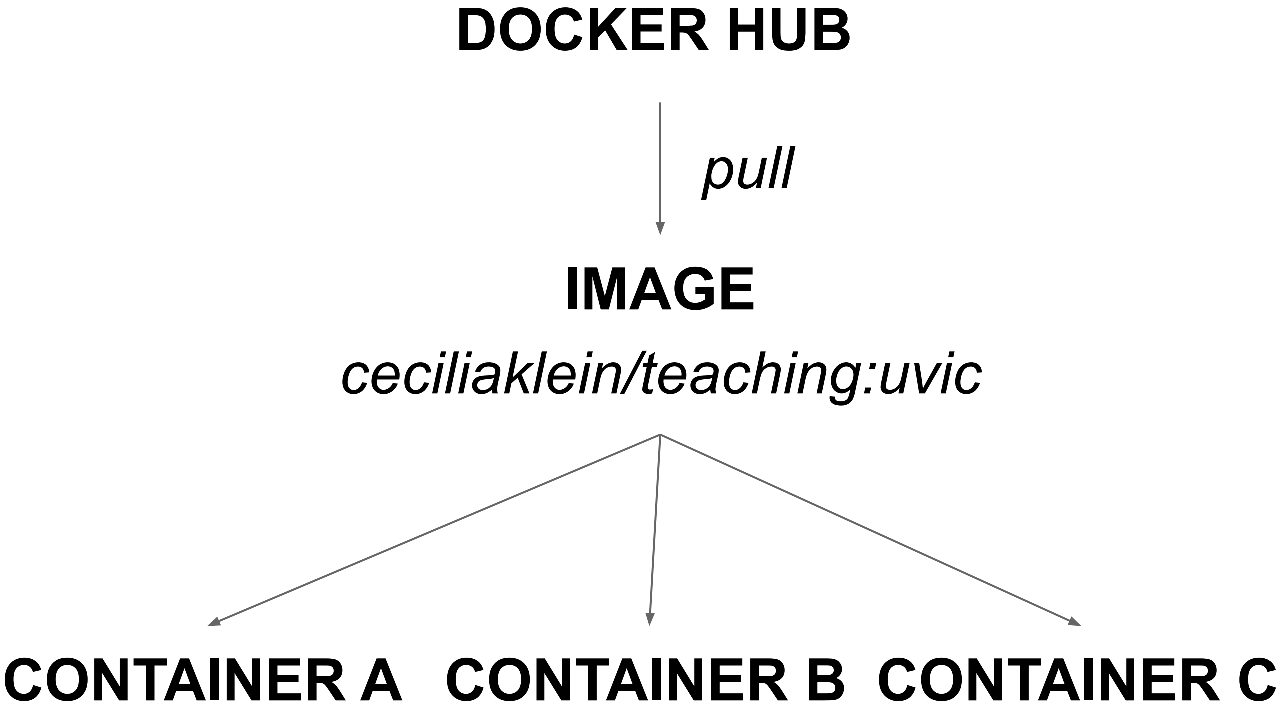
CONTAINERS

A container is a runtime instance of a docker image. A container will always run the same, regardless of the infrastructure.

DOCKER HUB

Docker Hub is a service provided by Docker for finding and sharing container images with your team. Learn more and find images at *https://hub.docker.com*

Basics Docker



DOCKER HUB

pull

IMAGE

ceciliaklein/teaching:uvic

CONTAINER A **CONTAINER B** **CONTAINER C**

Same results in all containers

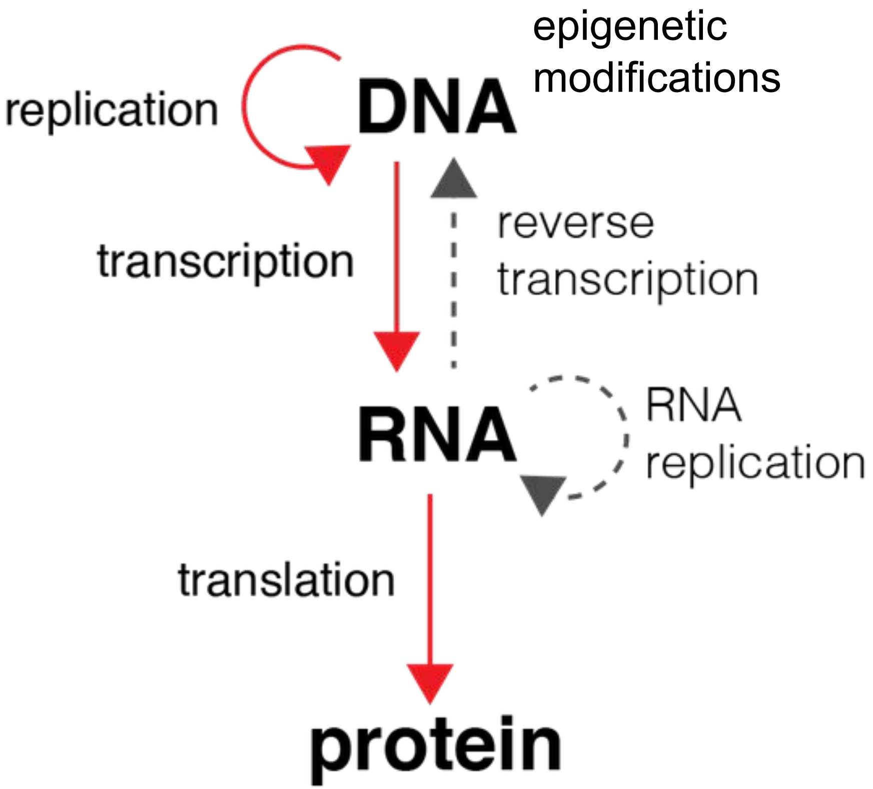
Hands-on

Basic concepts and setup 2.1 / 2.2

https://public-docs.crg.es/rguigo/Data/cklein/courses/UVIC/handsOn/#_basic_concepts_and_setup

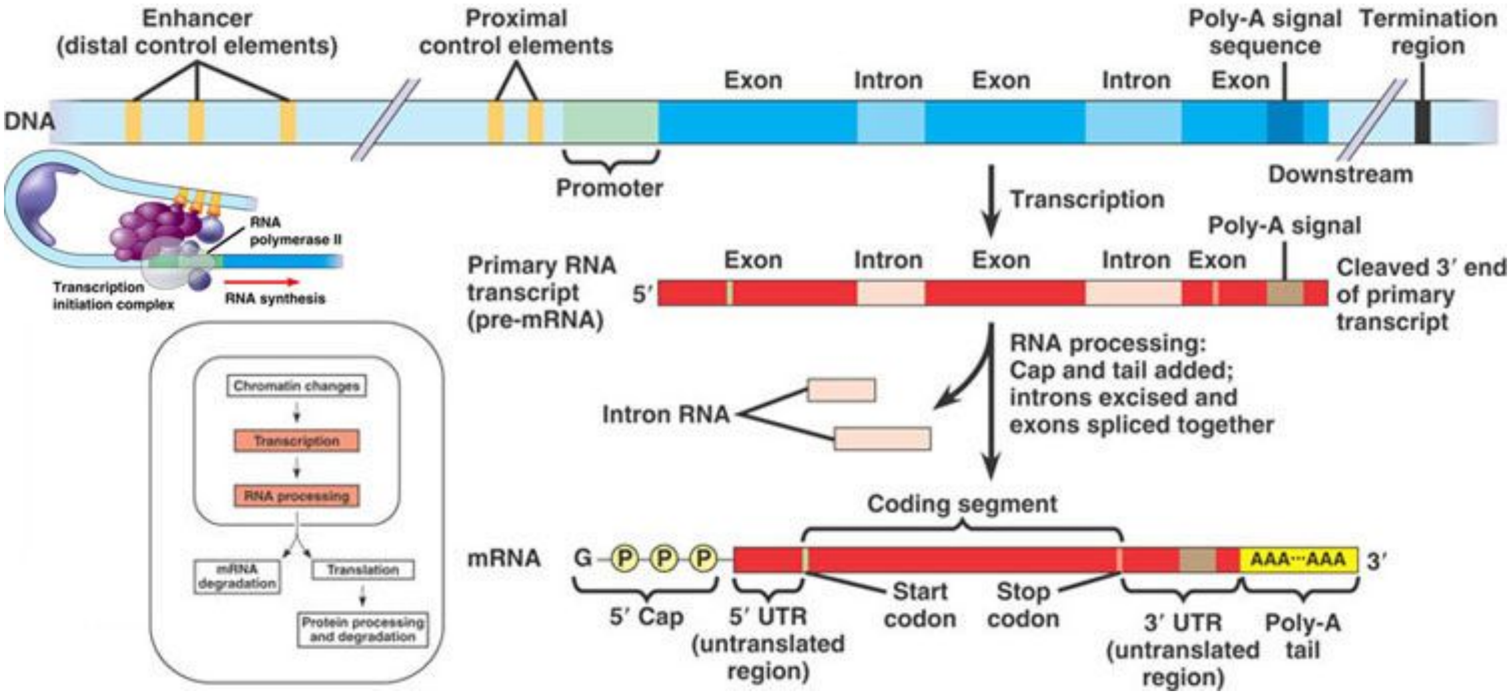
RNA biology

Molecular biology dogma



- Only ~1% of the human genome produces proteins, although much more is transcribed (~60%).
- The genome is identical in all cell types, however not all cell types have the same function. That's why the transcriptome (and the epigenome) becomes also relevant.

RNA transcription and processing



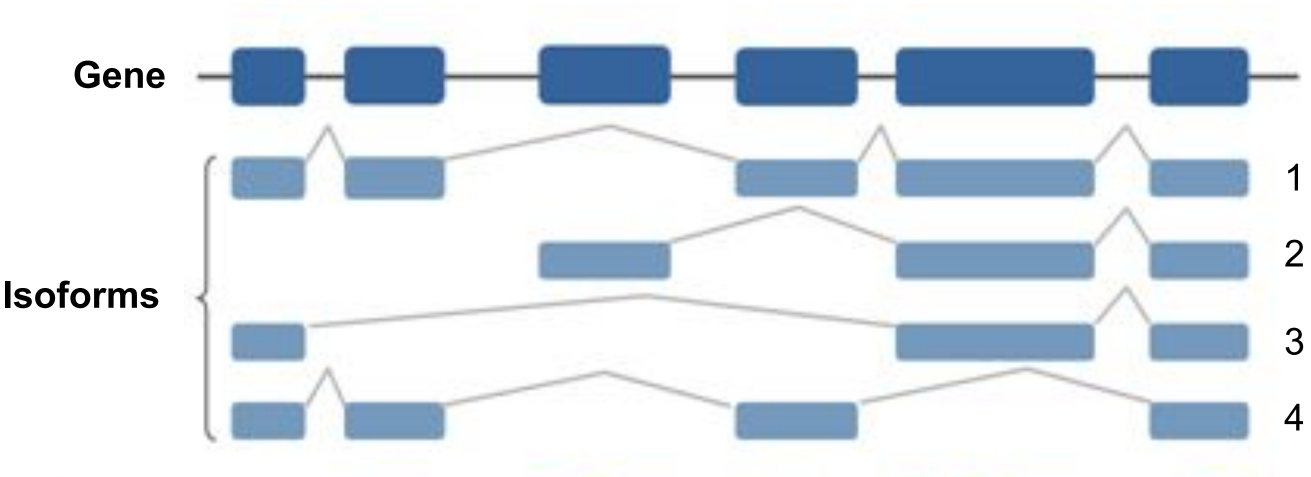
Primary RNA transcripts are extensively processed: capping, splicing, polyadenylation, editing

This process is highly regulated and results in a gene producing many distinct transcript isoforms: one gene, many transcripts

The transcriptome is distinct from and more complex than the genome

The transcriptome cannot be predicted from the genome sequence alone: it must be measured

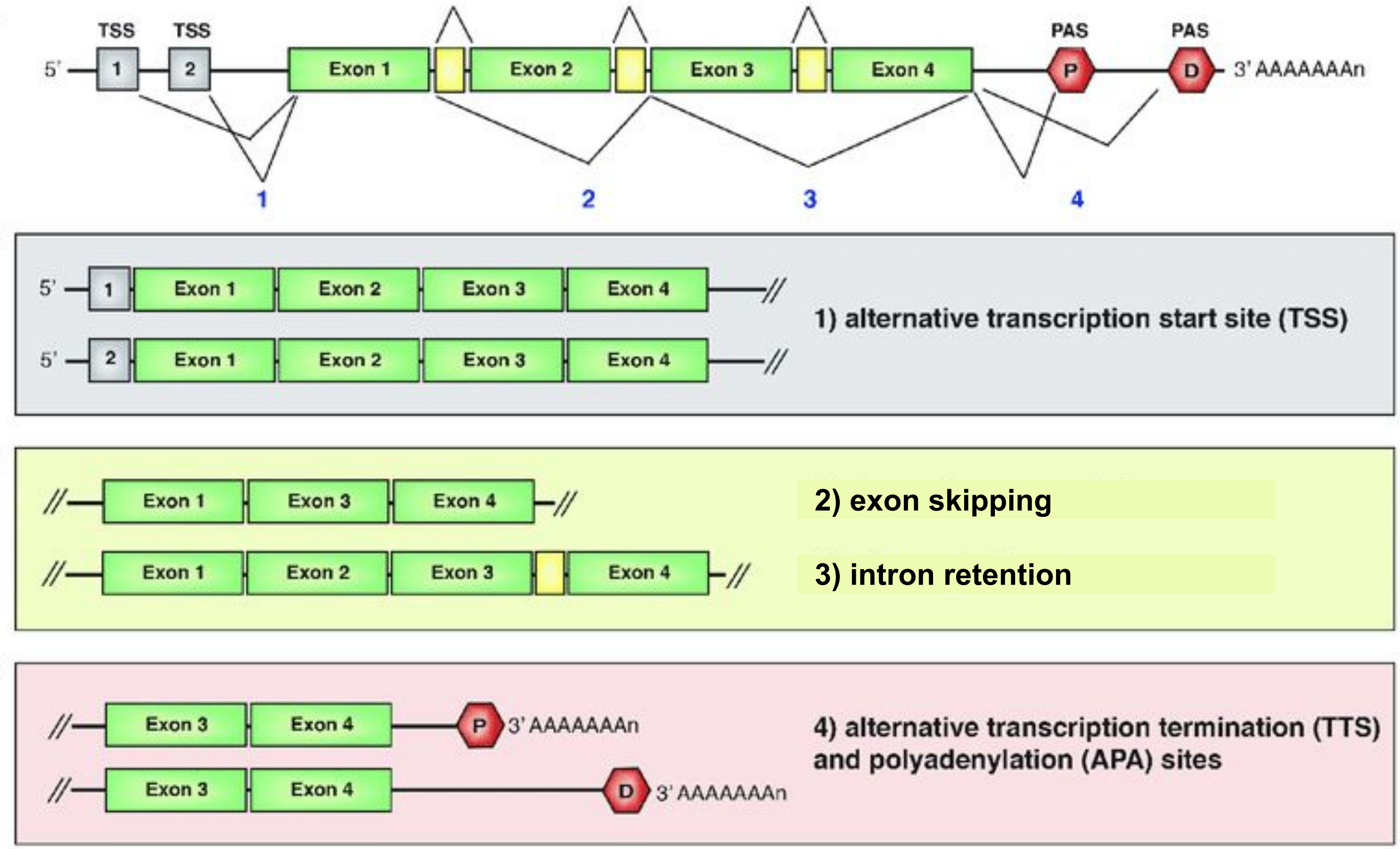
Genome and transcriptome



Some definitions:

- Genome: the full DNA complement of a species’ cell
- Gene: the physical region of a chromosome producing some kind or RNA transcript
- Isoforms: distinct RNAs arising from the gene, through differential exon inclusion, transcription start or termination sites.
- Transcript: The RNA molecule corresponding to one of the isoforms
- Transcriptome: the full RNA complement of a species' cell

Complexity arising from differential processing

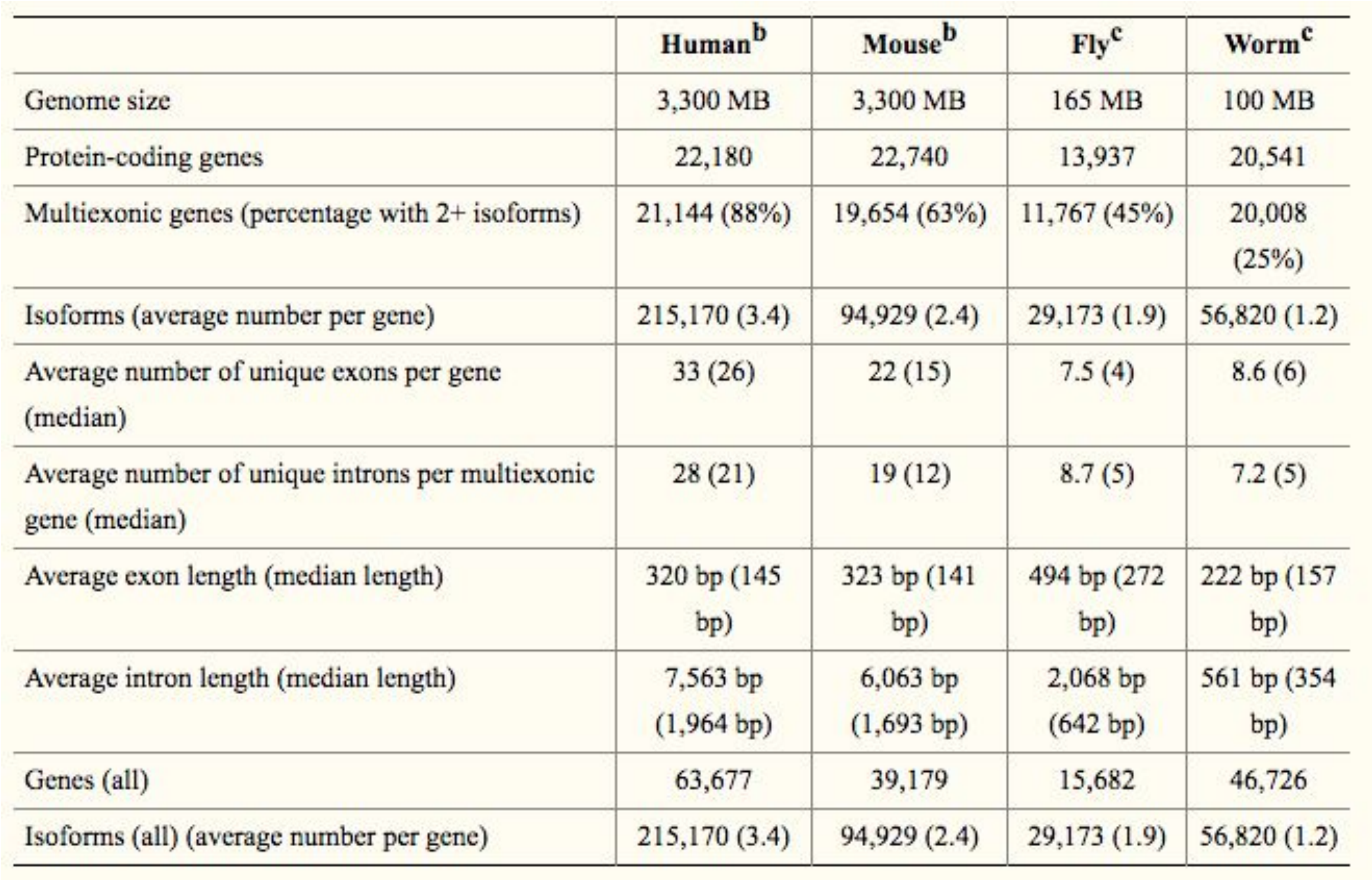

TSS
TSS
PAS
PAS
5'
1
2
Exon 1
Exon 2
Exon 3
Exon 4
P
D
3' AAAAAAAn
1
2
3
4
1) alternative transcription start site (TSS)
2) exon skipping
3) intron retention
4) alternative transcription termination (TTS) and polyadenylation (APA) sites

These processing events can result in different protein products, differentially (post-) transcriptionally regulated mRNAs or non-protein coding isoforms.

Cecilia Coimbra Klein

Andreassi, C. *et al.* (2018). doi: 10.3389/fnmol.2018.00304

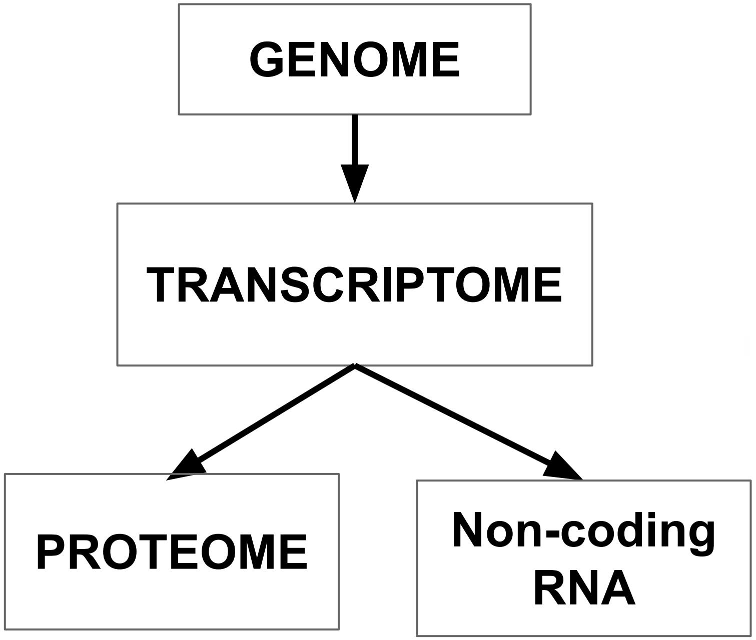
Complexity arising from differential processing



	Human[b]	Mouse[b]	Fly[c]	Worm[c]
Genome size	3,300 MB	3,300 MB	165 MB	100 MB
Protein-coding genes	22,180	22,740	13,937	20,541
Multiexonic genes (percentage with 2+ isoforms)	21,144 (88%)	19,654 (63%)	11,767 (45%)	20,008 (25%)
Isoforms (average number per gene)	215,170 (3.4)	94,929 (2.4)	29,173 (1.9)	56,820 (1.2)
Average number of unique exons per gene (median)	33 (26)	22 (15)	7.5 (4)	8.6 (6)
Average number of unique introns per multiexonic gene (median)	28 (21)	19 (12)	8.7 (5)	7.2 (5)
Average exon length (median length)	320 bp (145 bp)	323 bp (141 bp)	494 bp (272 bp)	222 bp (157 bp)
Average intron length (median length)	7,563 bp (1,964 bp)	6,063 bp (1,693 bp)	2,068 bp (642 bp)	561 bp (354 bp)
Genes (all)	63,677	39,179	15,682	46,726
Isoforms (all) (average number per gene)	215,170 (3.4)	94,929 (2.4)	29,173 (1.9)	56,820 (1.2)

Lee & Rio (2015). doi:10.1146/annurev-biochem-060614-034316

RNA composition in the cell

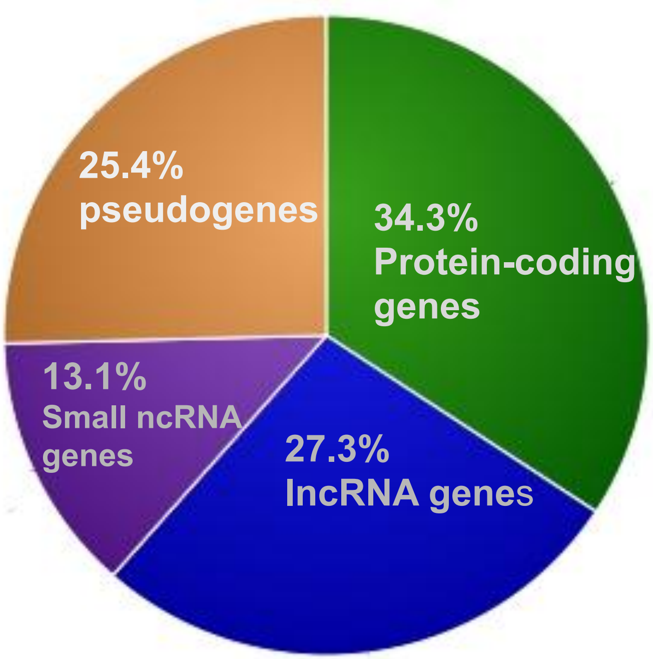


GENOME

TRANSCRIPTOME

PROTEOME

Non-coding RNA



From gencode v.26 annotation

- Only part of the human transcriptome encode proteins
- Many different type of regulatory RNAs, small <200nt and long >200nt
- lncRNAs: transcribed by RNA Polymerase II, actively processed
- Functionally important, have many signatures of mRNAs
- XIST, HOTAIR, TelRNAs

Reference gene annotation

Reference gene annotation

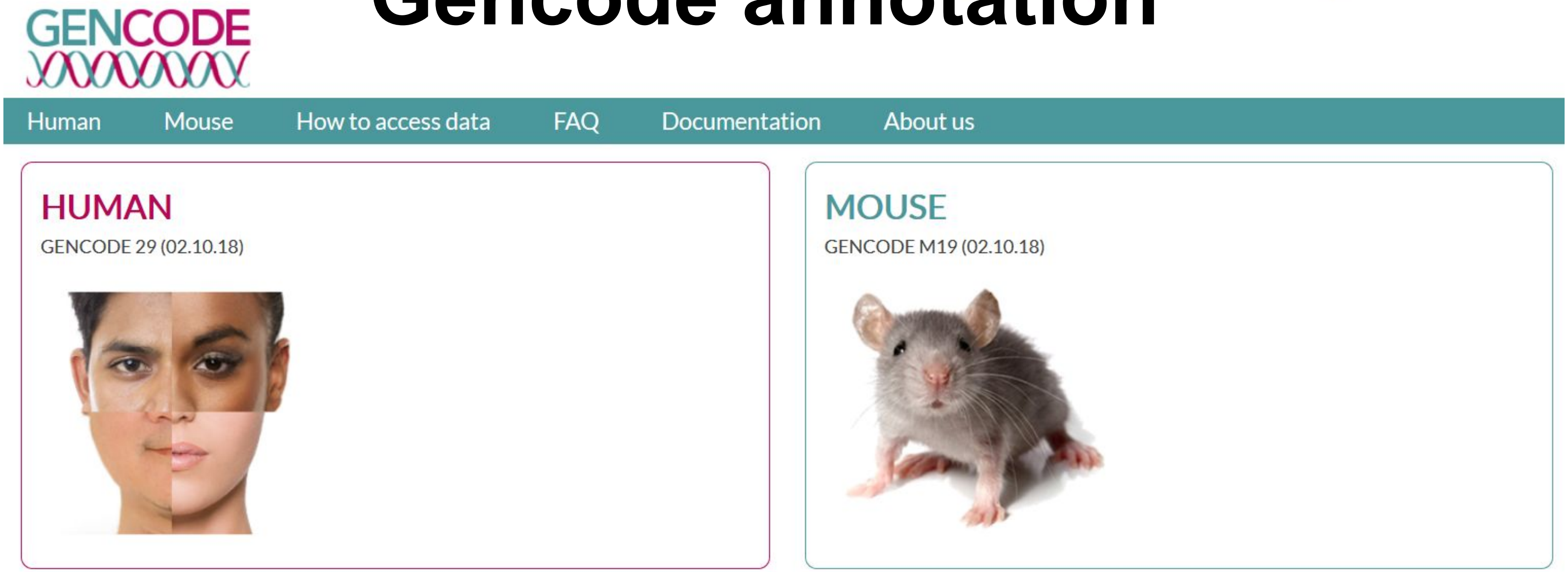
- For a given species and associated genome assembly, the reference gene annotation is the collection of all genes known for this species

- A gene annotation (like a genome assembly) can be at various completion stages depending on the species. High-quality annotations: human, mouse, *D. melanogaster*, *C. elegans* or yeast.

- It is important to choose well the reference gene annotation beforehand since it will represent the known transcriptome to which the RNA-seq transcriptome will be compared.

Always check the annotation version you’re going to use.

Gencode annotation



https://www.gencodegenes.org/

- **4 broad gene categories**: protein-coding genes (~20,000), long non-coding genes, pseudogenes, small non-coding genes
- **Several features:** gene, transcript, exon, CDS, UTR
- **3 confidence levels**: automatically annotated < manually annotated < validated
- **File formats**: GTF/GFF3

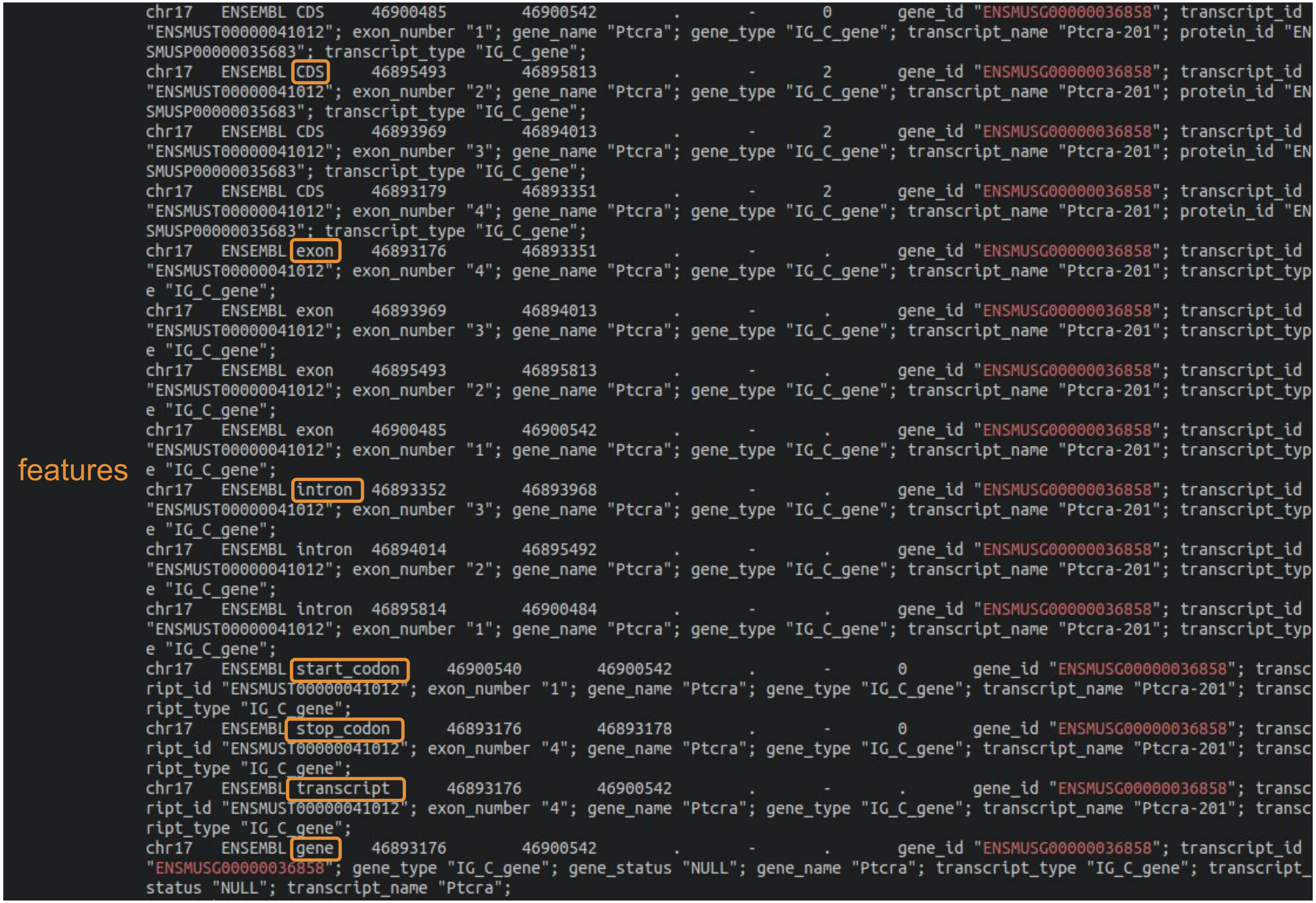
Gencode lncRNA gene annotation

- Gencode has always annotated lncRNA genes and was calling them “processed_transcript”
- Since they are more and more numerous and interesting to people, Gencode now better classifies them, partly using their location to PCGs:

3prime_overlapping_ncrna	Transcripts where ditag and/or published experimental data strongly supports the existence of long non-coding transcripts transcribed from the 3'UTR.
sense_intronic	Long non-coding transcript in introns of a coding gene that does not overlap any exons.
sense_overlapping	Long non-coding transcript that contains a coding gene in its intron on the same strand.
antisense	Transcript believed to be an antisense product used in the regulation of the gene to which it belongs.
non_coding	Transcript which is known from the literature to not be protein coding.
processed_transcript	Doesn't contain an ORF.
lincRNA	Long, intervening noncoding (linc)RNAs, that can be found in evolutionarily conserved, intergenic regions.

GTF format

a text-based format for storing features information



features

```
chr17   ENSEMBL CDS     46900485        46900542        .       -       0       gene_id "ENSMUSG00000036858"; transcript_id
"ENSMUST00000041012"; exon_number "1"; gene_name "Ptcra"; gene_type "IG_C_gene"; transcript_name "Ptcra-201"; protein_id "EN
SMUSP00000035683"; transcript_type "IG_C_gene";
chr17   ENSEMBL CDS     46895493        46895813        .       -       2       gene_id "ENSMUSG00000036858"; transcript_id
"ENSMUST00000041012"; exon_number "2"; gene_name "Ptcra"; gene_type "IG_C_gene"; transcript_name "Ptcra-201"; protein_id "EN
SMUSP00000035683"; transcript_type "IG_C_gene";
chr17   ENSEMBL CDS     46893969        46894013        .       -       2       gene_id "ENSMUSG00000036858"; transcript_id
"ENSMUST00000041012"; exon_number "3"; gene_name "Ptcra"; gene_type "IG_C_gene"; transcript_name "Ptcra-201"; protein_id "EN
SMUSP00000035683"; transcript_type "IG_C_gene";
chr17   ENSEMBL CDS     46893179        46893351        .       -       2       gene_id "ENSMUSG00000036858"; transcript_id
"ENSMUST00000041012"; exon_number "4"; gene_name "Ptcra"; gene_type "IG_C_gene"; transcript_name "Ptcra-201"; protein_id "EN
SMUSP00000035683"; transcript_type "IG_C_gene";
chr17   ENSEMBL exon    46893176        46893351        .       -       .       gene_id "ENSMUSG00000036858"; transcript_id
"ENSMUST00000041012"; exon_number "4"; gene_name "Ptcra"; gene_type "IG_C_gene"; transcript_name "Ptcra-201"; transcript_typ
e "IG_C_gene";
chr17   ENSEMBL exon    46893969        46894013        .       -       .       gene_id "ENSMUSG00000036858"; transcript_id
"ENSMUST00000041012"; exon_number "3"; gene_name "Ptcra"; gene_type "IG_C_gene"; transcript_name "Ptcra-201"; transcript_typ
e "IG_C_gene";
chr17   ENSEMBL exon    46895493        46895813        .       -       .       gene_id "ENSMUSG00000036858"; transcript_id
"ENSMUST00000041012"; exon_number "2"; gene_name "Ptcra"; gene_type "IG_C_gene"; transcript_name "Ptcra-201"; transcript_typ
e "IG_C_gene";
chr17   ENSEMBL exon    46900485        46900542        .       -       .       gene_id "ENSMUSG00000036858"; transcript_id
"ENSMUST00000041012"; exon_number "1"; gene_name "Ptcra"; gene_type "IG_C_gene"; transcript_name "Ptcra-201"; transcript_typ
e "IG_C_gene";
chr17   ENSEMBL intron  46893352        46893968        .       -       .       gene_id "ENSMUSG00000036858"; transcript_id
"ENSMUST00000041012"; exon_number "3"; gene_name "Ptcra"; gene_type "IG_C_gene"; transcript_name "Ptcra-201"; transcript_typ
e "IG_C_gene";
chr17   ENSEMBL intron  46894014        46895492        .       -       .       gene_id "ENSMUSG00000036858"; transcript_id
"ENSMUST00000041012"; exon_number "2"; gene_name "Ptcra"; gene_type "IG_C_gene"; transcript_name "Ptcra-201"; transcript_typ
e "IG_C_gene";
chr17   ENSEMBL intron  46895814        46900484        .       -       .       gene_id "ENSMUSG00000036858"; transcript_id
"ENSMUST00000041012"; exon_number "1"; gene_name "Ptcra"; gene_type "IG_C_gene"; transcript_name "Ptcra-201"; transcript_typ
e "IG_C_gene";
chr17   ENSEMBL start_codon     46900540        46900542        .       -       0       gene_id "ENSMUSG00000036858"; transc
ript_id "ENSMUST00000041012"; exon_number "1"; gene_name "Ptcra"; gene_type "IG_C_gene"; transcript_name "Ptcra-201"; transc
ript_type "IG_C_gene";
chr17   ENSEMBL stop_codon      46893176        46893178        .       -       0       gene_id "ENSMUSG00000036858"; transc
ript_id "ENSMUST00000041012"; exon_number "4"; gene_name "Ptcra"; gene_type "IG_C_gene"; transcript_name "Ptcra-201"; transc
ript_type "IG_C_gene";
chr17   ENSEMBL transcript      46893176        46900542        .       -       .       gene_id "ENSMUSG00000036858"; transc
ript_id "ENSMUST00000041012"; exon_number "4"; gene_name "Ptcra"; gene_type "IG_C_gene"; transcript_name "Ptcra-201"; transc
ript_type "IG_C_gene";
chr17   ENSEMBL gene    46893176        46900542        .       -       .       gene_id "ENSMUSG00000036858"; transcript_id
"ENSMUSG00000036858"; gene_type "IG_C_gene"; gene_status "NULL"; gene_name "Ptcra"; transcript_type "IG_C_gene"; transcript_
status "NULL"; transcript_name "Ptcra";
```

Hands-on

Reference gene annotation 2.3

https://public-docs.crg.es/rguigo/Data/cklein/courses/UVIC/handsOn/#_reference_gene_annotation

Next generation sequencing

NGS: Illumina sequencing

- Illumina Sequencing (short reads ~ max. 150bp)

    - *single end*

        1) Library preparation: DNA fragmentation, adapter ligation, PCR amplification
        2) Solid-phase *bridge* amplification
        3) Flowing of fluorescent reversible terminator dNTPs; incorporation of a single base per cycle. *Sequencing by synthesis*.
        4) Read identity of each base of a cluster from sequential images

    - *paired end*

        5) After completion of the first read, the templates can be regenerated *in situ* to enable a second read from the opposite end.

NGS: Third generation sequencing

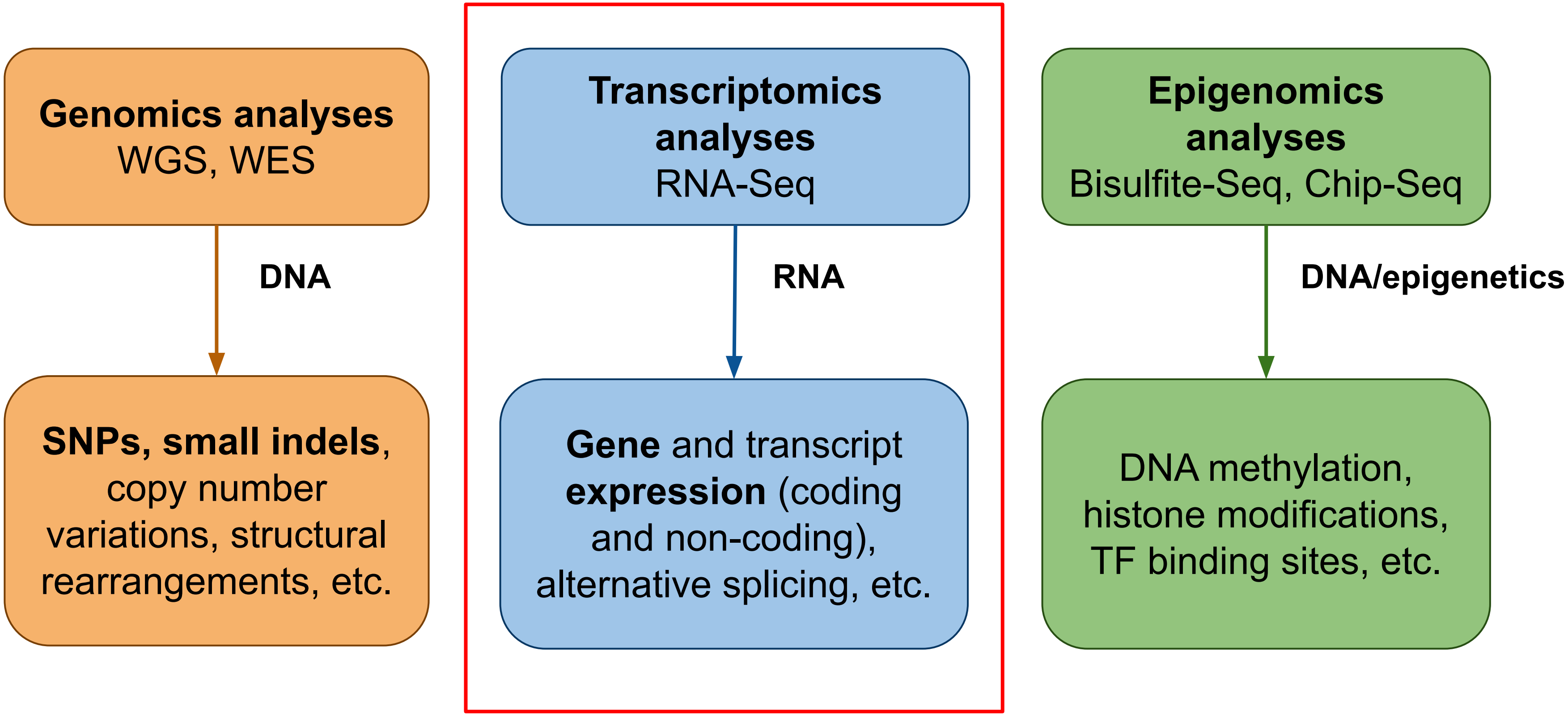
- Although Illumina is by far the most popular, there are many other sequencing technologies, such as PacBio, Ion Torrent or Oxford NanoPore that:

  - allow sequencing genomic material without neither fragmentation nor clonal amplification.

  - enable getting longer reads (tens of Kb!), but at the price of a much higher error rate than Illumina.

  - have been mostly used for genome sequencing, since those reads can span complicated repeat-rich regions which are trickier to assemble using short reads.

Which *-Seq do I need?


Genomics analyses
WGS, WES

DNA

SNPs, small indels, copy number variations, structural rearrangements, etc.

Transcriptomics analyses
RNA-Seq

RNA

Gene and transcript **expression** (coding and non-coding), alternative splicing, etc.

Epigenomics analyses
Bisulfite-Seq, Chip-Seq

DNA/epigenetics

DNA methylation, histone modifications, TF binding sites, etc.

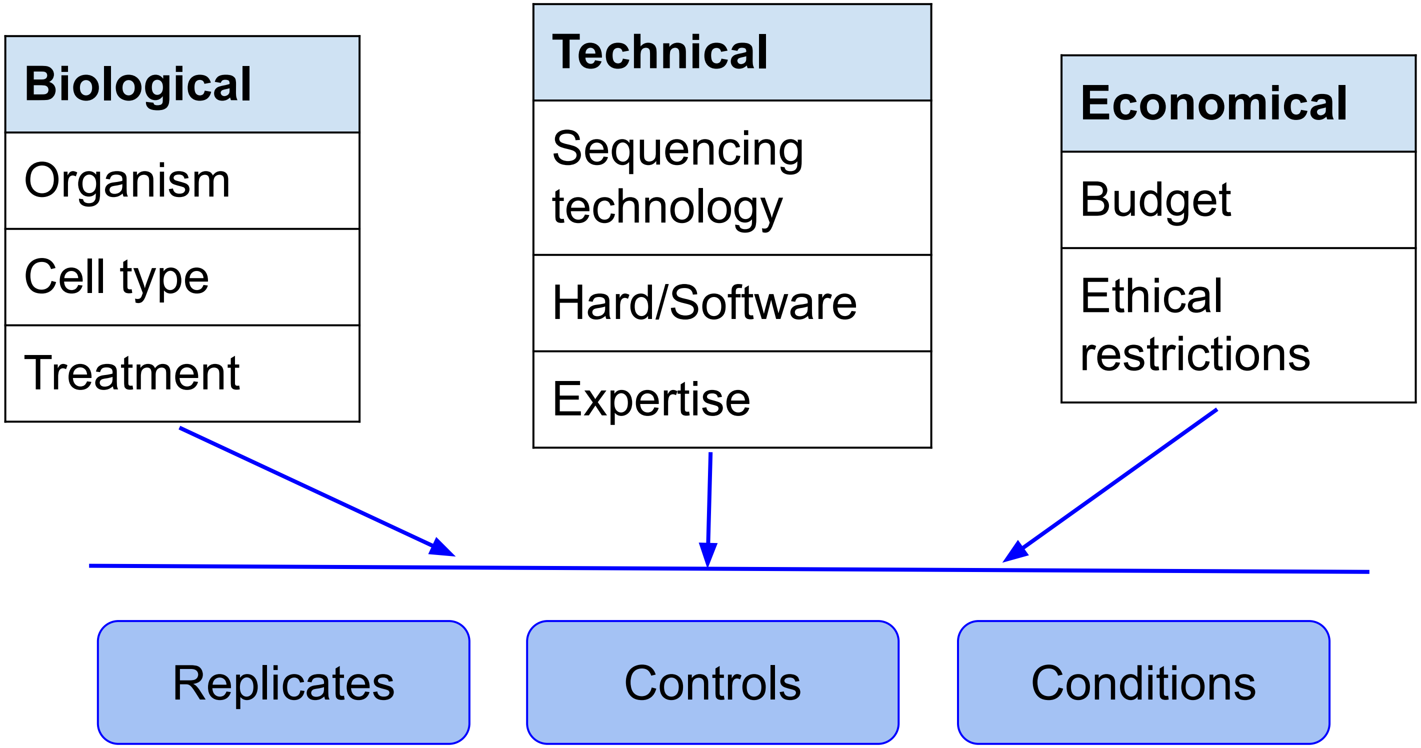
- Learn more about your favourite *-Seq here!
- Note that we are always talking about *re-sequencing*, which is something different from *de novo sequencing* (what is done for a new genome assembly)

RNA sequencing

Why is it useful?

- Measure gene and transcript expression at different conditions, developmental stages, etc.
- Discover / annotate novel elements: genes (coding and non-coding), transcripts, exons, (chimeric) junctions, circular RNAs, etc.
- Alternative splicing, transcription start and termination (polyadenylation) sites.

Experimental design



Biological

Organism

Cell type

Treatment

Technical

Sequencing technology

Hard/Software

Expertise

Economical

Budget

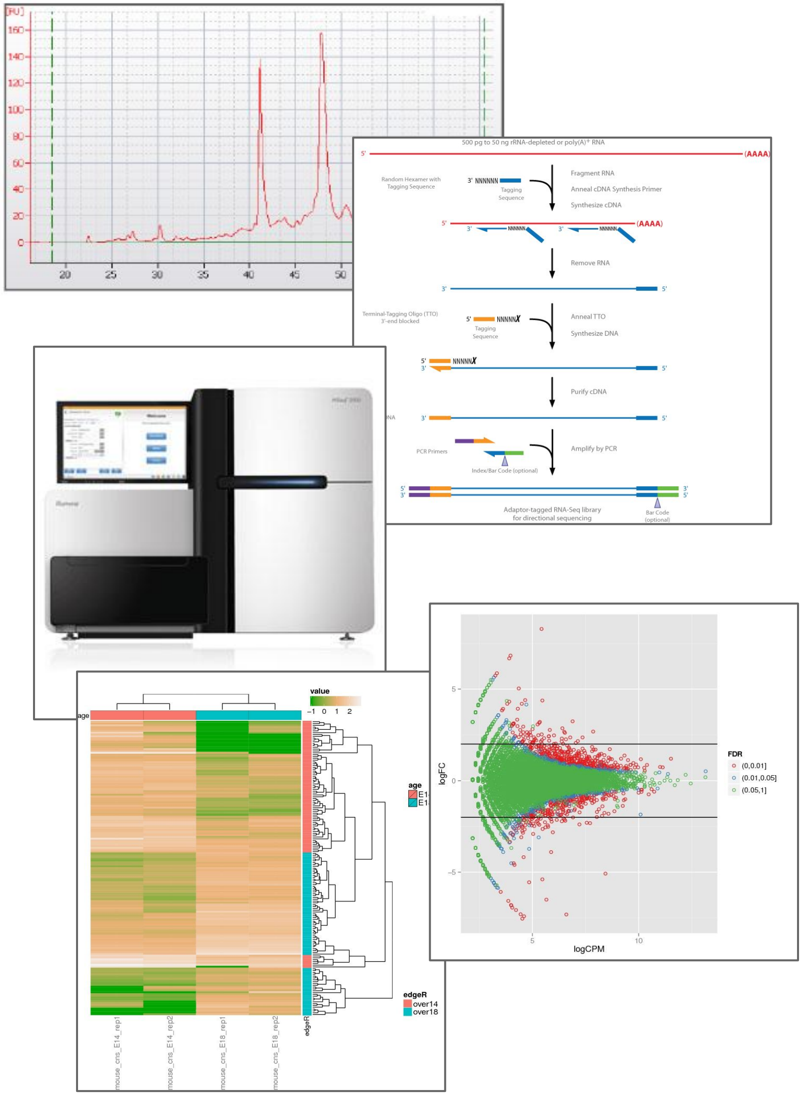
Ethical restrictions

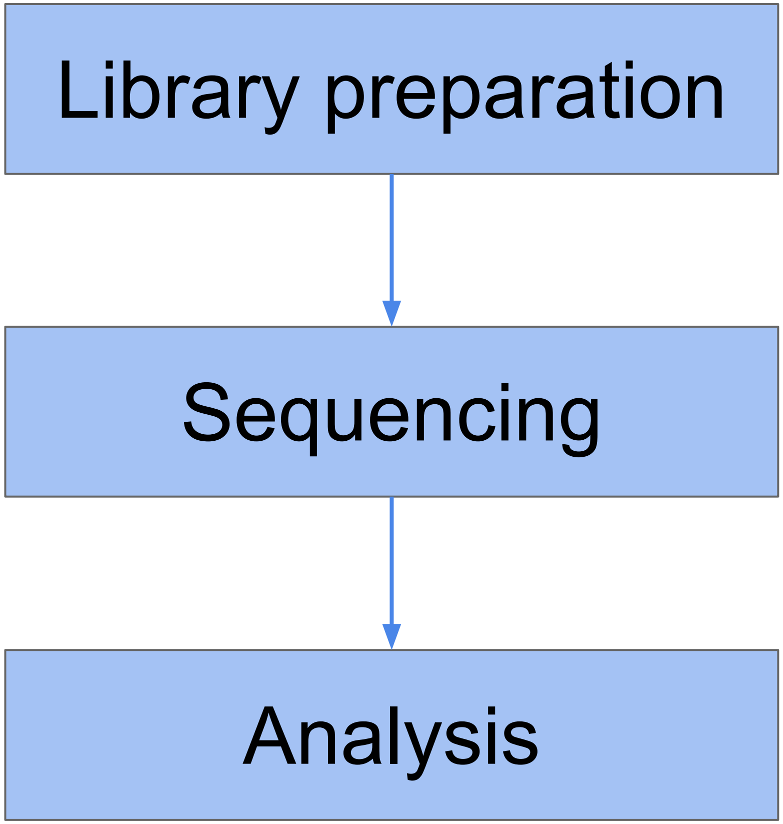
Replicates

Controls

Conditions

RNA-seq experiment





Library preparation

Sequencing

Analysis

Cecilia Coimbra Klein

Experimental variables of RNA-seq

Cellular localization
Whole cell
Chromatin
Exosome
Nucleus
Cytoplasm

RNA purification
Total RNA
PolyA+
PolyA-
Ribo-

Size selection
Long (>200nt)
Short (<200nt)

Preparation
Single end
Paired end

Strandness
Stranded
Unstranded

Special protocols
Single-cell RNA-seq
Nascent RNA-seq (GRO-seq/NUN-seq)
miRNA-seq

Experimental variables of RNA-seq

Cellular localization
Whole cell
Chromatin
Exosome
Nucleus
Cytoplasm

RNA purification
Total RNA
PolyA+
PolyA-
Ribo-

Size selection
Long (>200nt)
Short (<200nt)

Preparation
Single end
Paired end

Strandness
Stranded
Unstranded

Special protocols
Single-cell RNA-seq
Nascent RNA-seq (GRO-seq/NUN-seq)
miRNA-seq

Experimental variables of RNA-seq

Cellular localization
Whole cell
Chromatin
Exosome
Nucleus
Cytoplasm

RNA purification
Total RNA
PolyA+
PolyA-
Ribo-

Size selection
Long (>200nt)
Short (<200nt)

Preparation
Single end
Paired end

Strandness
Stranded
Unstranded

Special protocols
Single-cell RNA-seq
Nascent RNA-seq (GRO-seq/NUN-seq)
miRNA-seq

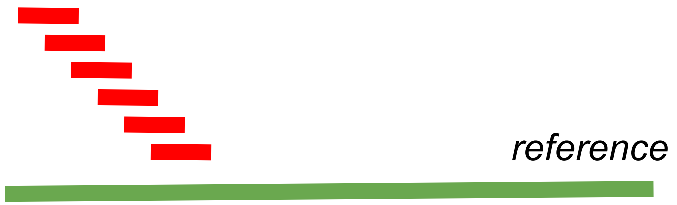
OUR HANDS-ON

RNA purification protocol

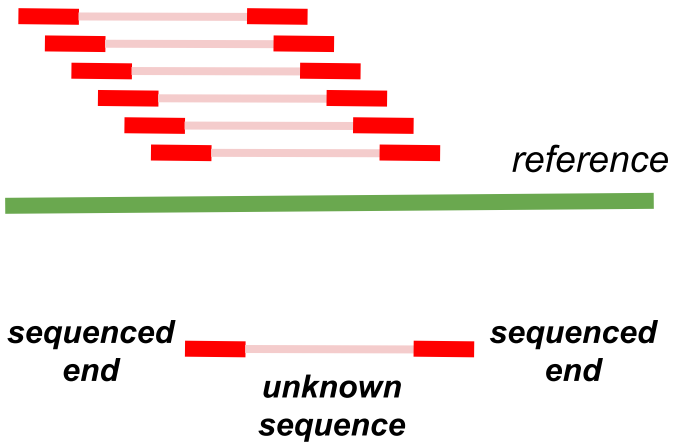
- **PolyA+** gets rid of the ribosomal RNAs and purify mature polyadenylated transcripts.
- **PolyA-** enrichs for non-mature RNAs
- **Ribo-** gets rid of the ribosomal RNAs but capture both mature and non-mature RNAs

Preparation

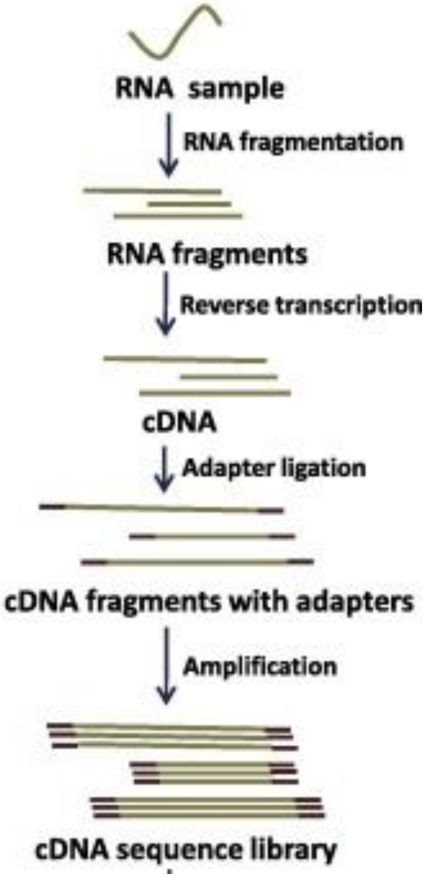
Single-end (SE) reads



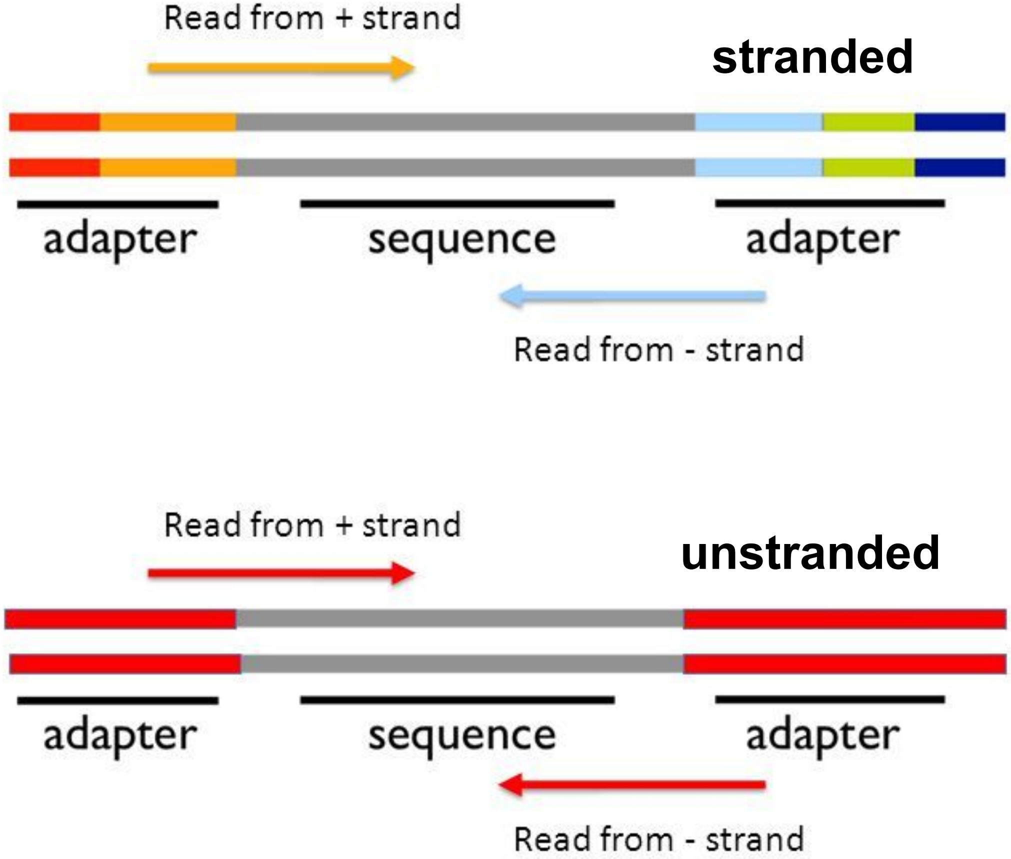
Paired-end (PE) reads



Library preparation



Strandness



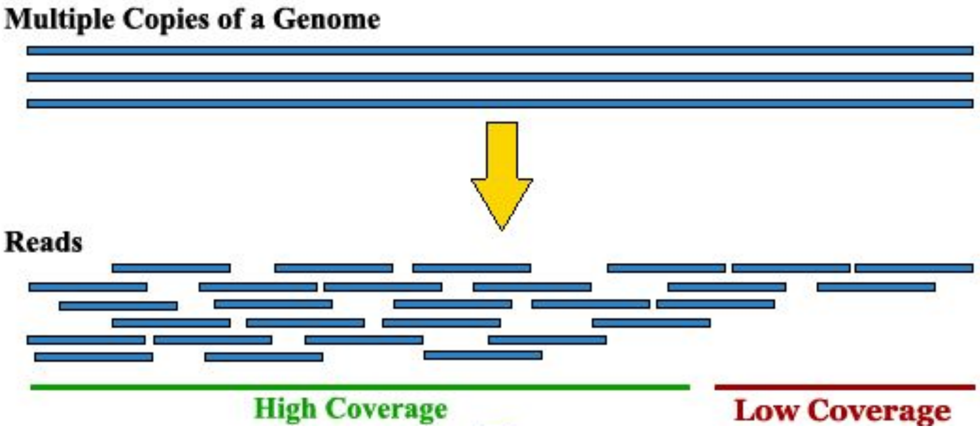
How much to sequence?

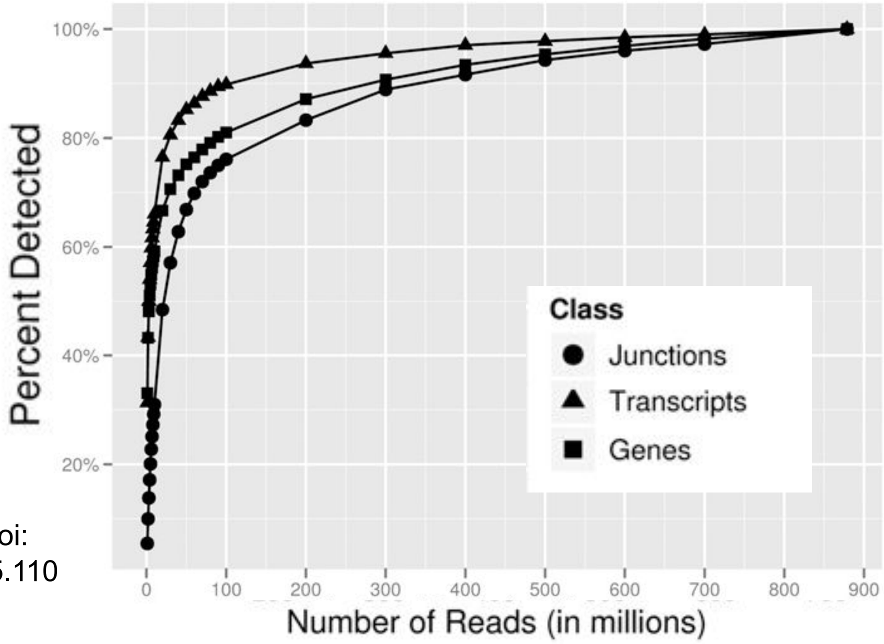
Depends on multiple factors:

- goal of experiment
- protocol
- species
- etc.

e.g. in humans:

>30M reads for simple analyses
>100M reads for novel elements discovery

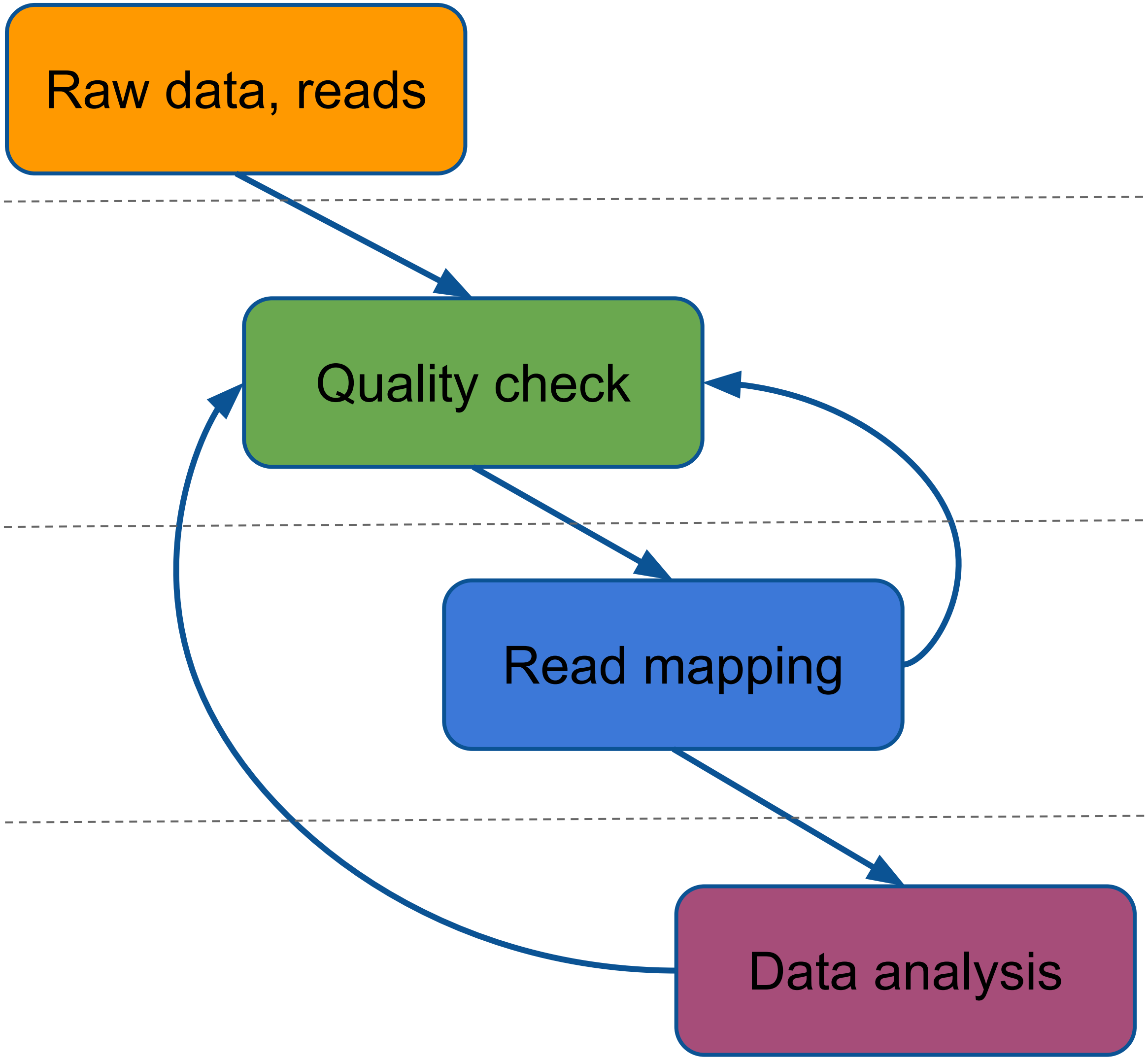




Toung, J. (2011) doi: 10.1101/gr.116335.110

Data formats

Typical pipeline



Raw data, reads

Quality check

Read mapping

Data analysis

Some data formats

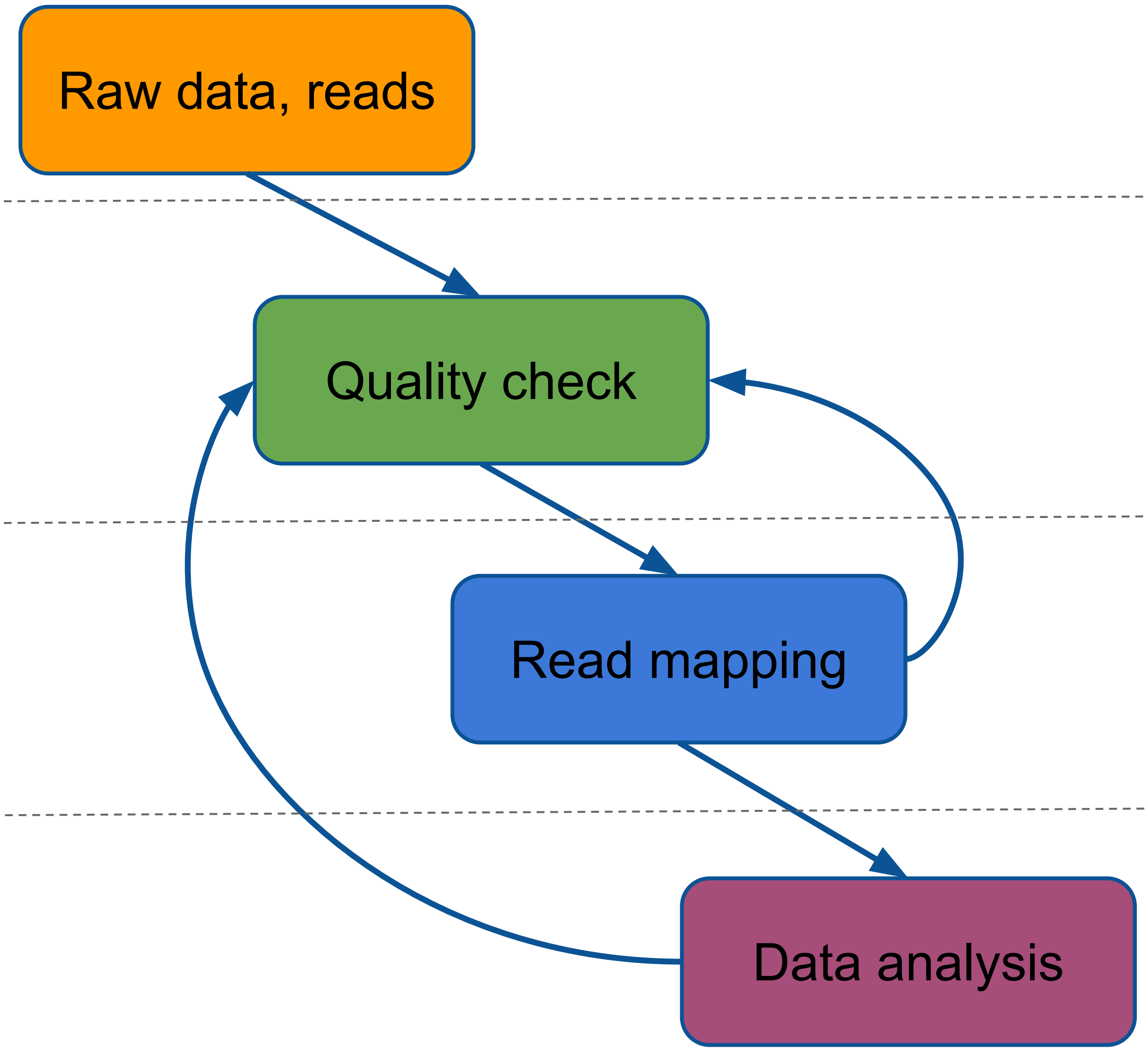
*.fastq, *.fa,
*.sff, *.sra

*.fastq
*.tsv, *.html..

*.sam, *.bam
*.bed, *.wig, *.bw
*.bedgraph
*.gtf, *.fa,..

*.vcf
*.tsv
*.ace, *.agp

Typical pipeline



Some data formats

*.fastq, *.fa,
*.sff, *.sra

*.fastq
*.tsv, *.html..

*.sam, *.bam
*.bed, *.wig, *.bw
*.bedgraph
*.gtf, *.fa,..

*.vcf
*.tsv
*.ace, *.agp

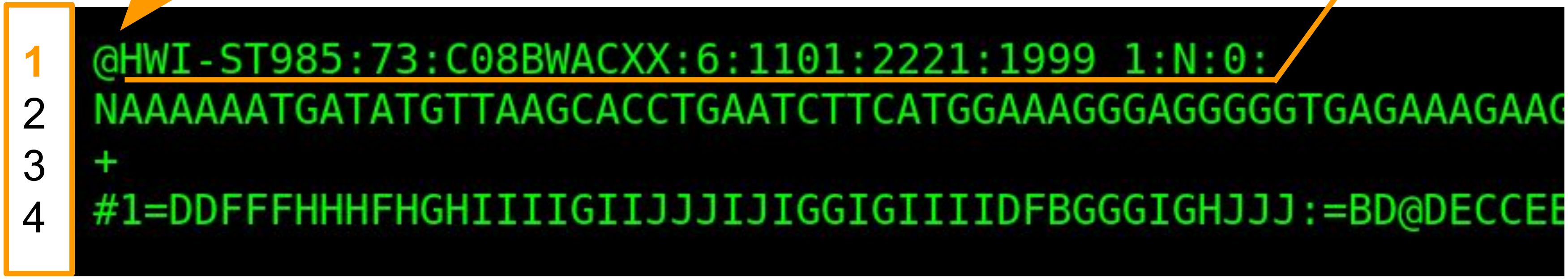
FASTQ format

FASTQ Format

a text-based format for storing biological sequences and their corresponding quality scores

1st character

Sequence id



```
@HWI-ST985:73:C08BWACXX:6:1101:2221:1999 1:N:0:
NAAAAAATGATATGTTAAGCACCTGAATCTTCATGGAAAGGGAGGGGGTGAGAAAGAAG
+
#1=DDFFFHHHFHGHIIIIGIIJJJIJIGGIGIIIIDFBGGGIGHJJJ:=BD@DECCEE
```

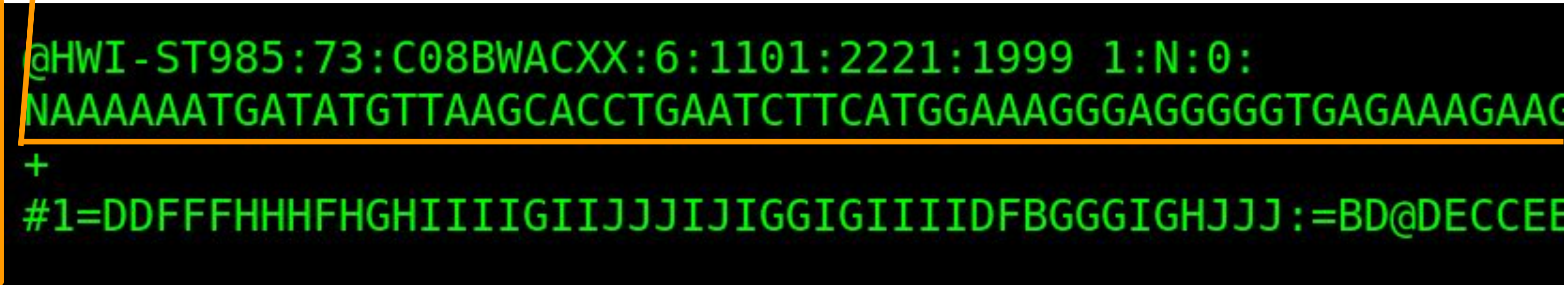
Optionally: The sequence id can be followed by a description

FASTQ Format

a text-based format for storing biological sequences and their corresponding quality scores

Raw sequence

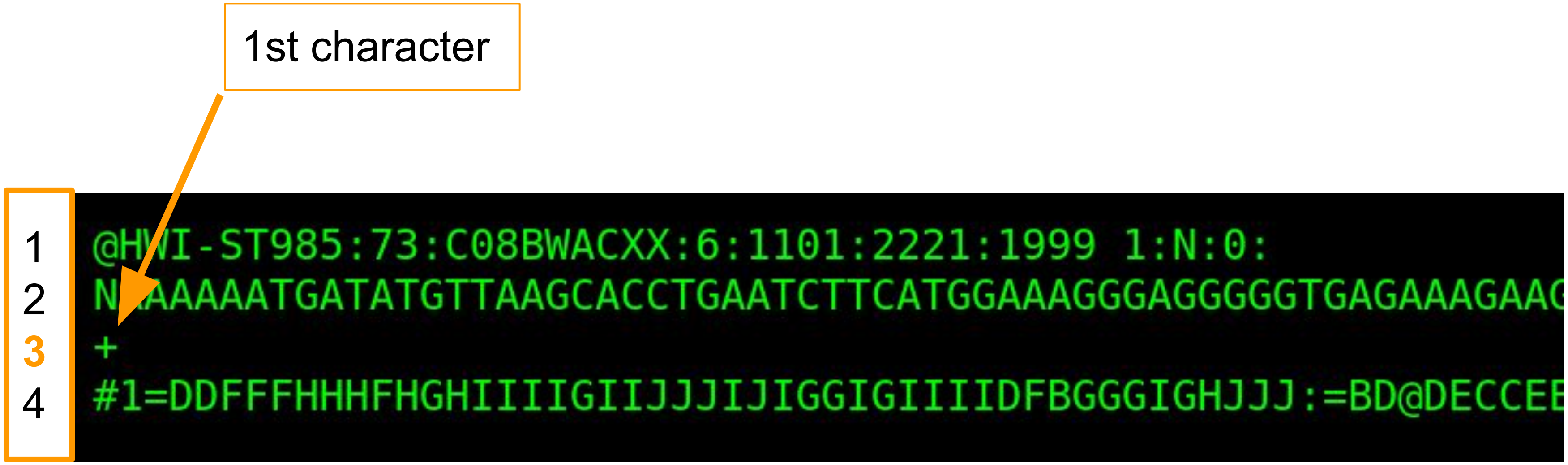
1
2
3
4



```
@HWI-ST985:73:C08BWACXX:6:1101:2221:1999 1:N:0:
NAAAAAATGATATGTTAAGCACCTGAATCTTCATGGAAAGGGAGGGGGTGAGAAAGAAG
+
#1=DDFFFHHHFHGHIIIIGIIJJJIJIGGIGIIIIDFBGGGIGHJJJ:=BD@DECCEE
```

FASTQ Format

a text-based format for storing biological sequences and their corresponding quality scores



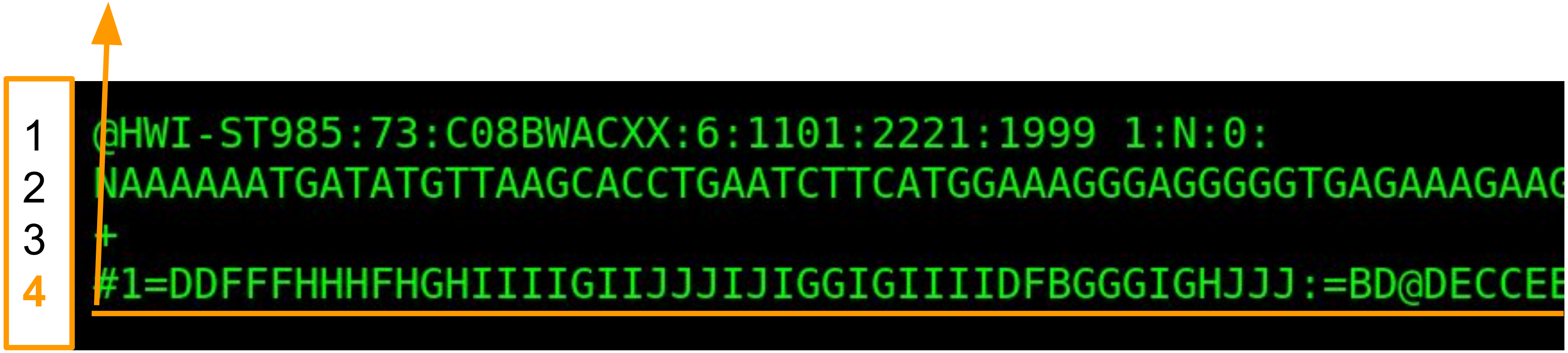
```
@HWI-ST985:73:C08BWACXX:6:1101:2221:1999 1:N:0:
NAAAAAATGATATGTTAAGCACCTGAATCTTCATGGAAAGGGAGGGGGTGAGAAAGAAG
+
#1=DDFFFHHHFHGHIIIIGIIJJJIJIGGIGIIIIDFBGGGIGHJJJ:=BD@DECCEE
```

Optionally: “+” can can be followed by the sequence id and any description

FASTQ Format

a text-based format for storing biological sequences and their corresponding quality scores

Quality code associated to each base of the sequence



```
@HWI-ST985:73:C08BWACXX:6:1101:2221:1999 1:N:0:
NAAAAAATGATATGTTAAGCACCTGAATCTTCATGGAAAGGGAGGGGGTGAGAAAGAAG
+
#1=DDFFFHHHFHGHIIIIGIIJJJIJIGGIGIIIIDFBGGGIGHJJJ:=BD@DECCEE
```

FASTQ Format - summary

Four lines per sequence are used in a FASTQ file:

1. begins with a '@' character and is followed by a sequence identifier and an *optional* description (like a FASTA title line)
2. the raw sequence
3. begins with a '+' character and is *optionally* followed by the same sequence identifier (and any description)
4. encodes the quality values for the sequence contained in line 2 (must contain the same number of symbols as the sequence)

FASTQ Format - quality offset

A quality value *Q* is an integer mapping of *p* (i.e., the probability that the corresponding base call is incorrect). The most used formula is the Phred quality score:

$$Q_{phred} = -10 \log_{10} p$$

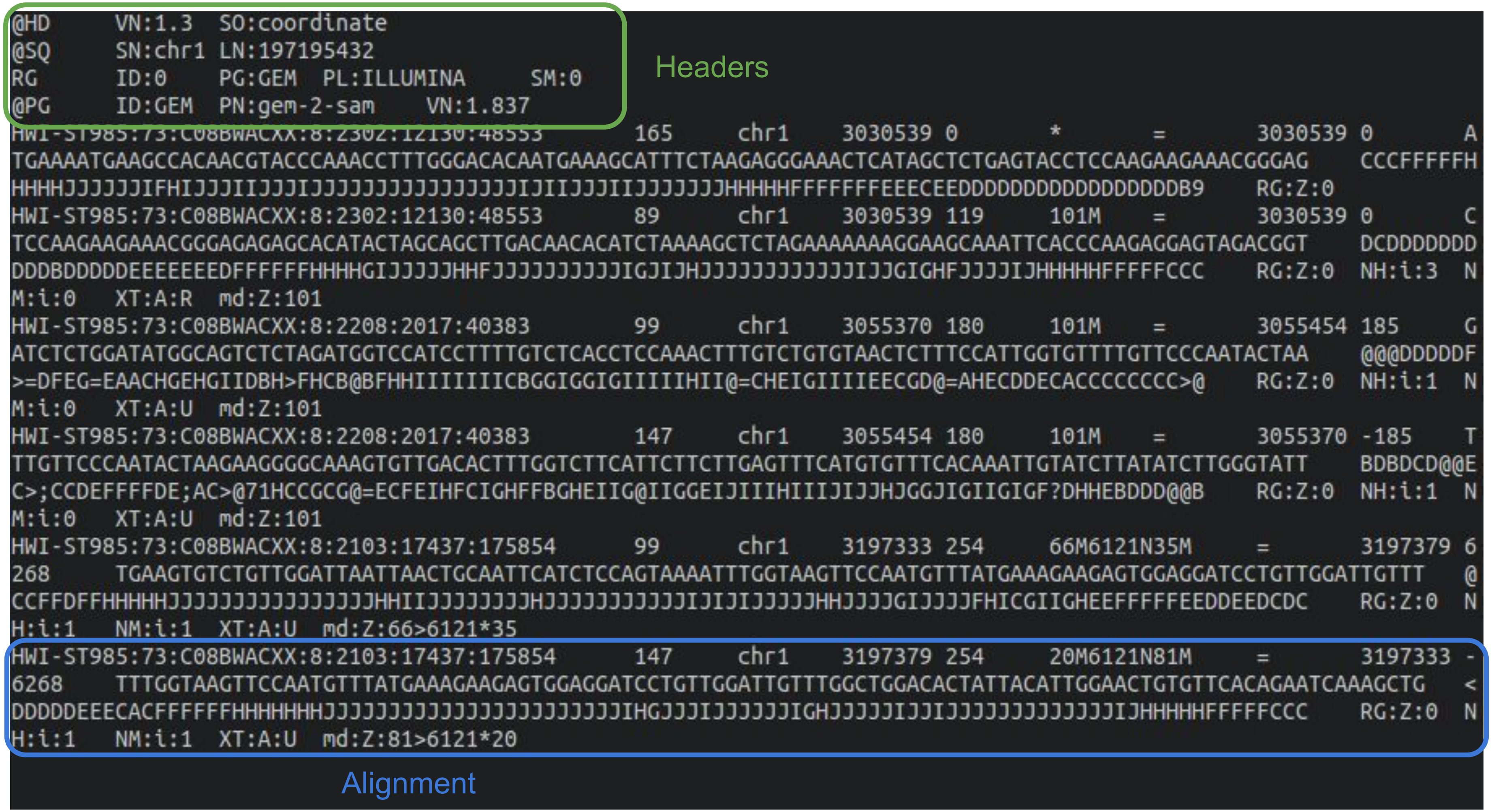
offset	max Phred score range	max ASCII range	real-world Phred score range	real-world ASCII range
33	0 - 93	33 - 126	0 - 40	33 - 73
64	0 - 62	64 - 126	0 - 40	64 - 104

https://en.wikipedia.org/wiki/FASTQ_format#Encoding

Cecilia Coimbra Klein

SAM format

Sequence Alignment/Map

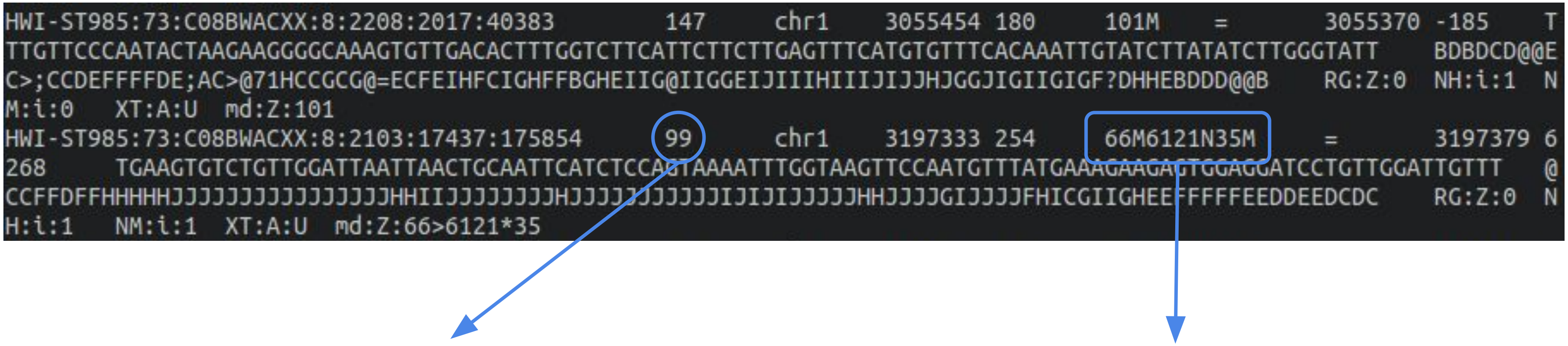


```
@HD     VN:1.3  SO:coordinate
@SQ     SN:chr1 LN:197195432
RG      ID:0    PG:GEM  PL:ILLUMINA     SM:0
@PG     ID:GEM  PN:gem-2-sam    VN:1.837
HWI-ST985:73:C08BWACXX:8:2302:12130:48553       165     chr1    3030539 0       *       =       3030539 0       A
TGAAAATGAAGCCACAACGTACCCAAACCTTTGGGACACAATGAAAGCATTTCTAAGAGGGAAACTCATAGCTCTGAGTACCTCCAAGAAGAAACGGGAG    CCCFFFFFH
HHHHJJJJJJIFHIJJJIIJJJIJJJJJJJJJJJJJJJJJJIJIIJJJIIJJJJJJJHHHHHFFFFFFEEECEEDDDDDDDDDDDDDDDDDB9    RG:Z:0
HWI-ST985:73:C08BWACXX:8:2302:12130:48553       89      chr1    3030539 119     101M    =       3030539 0       C
TCCAAGAAGAAACGGGAGAGAGCACATACTAGCAGCTTGACAACACATCTAAAAGCTCTAGAAAAAAAGGAAGCAAATTCACCCAAGAGGAGTAGACGGT    DCDDDDDDD
DDDBDDDDDEEEEEEEDFFFFFFHHHHGIJJJJJHHFJJJJJJJJJJJIGJIJHJJJJJJJJJJJJJIJJGIGHFJJJJIJHHHHHFFFFFCCC    RG:Z:0  NH:i:3  N
M:i:0   XT:A:R  md:Z:101
HWI-ST985:73:C08BWACXX:8:2208:2017:40383        99      chr1    3055370 180     101M    =       3055454 185     G
ATCTCTGGATATGGCAGTCTCTAGATGGTCCATCCTTTTGTCTCACCTCCAAACTTTGTCTGTGTAACTCTTTCCATTGGTGTTTTGTTCCCAATACTAA    @@@DDDDDF
>=DFEG=EAACHGEHGIIDBH>FHCB@BFHHIIIIIIIICBGGIGGIGIIIIIHII@=CHEIGIIIIEECGD@=AHECDDECACCCCCCCC>@    RG:Z:0  NH:i:1  N
M:i:0   XT:A:U  md:Z:101
HWI-ST985:73:C08BWACXX:8:2208:2017:40383        147     chr1    3055454 180     101M    =       3055370 -185    T
TTGTTCCCAATACTAAGAAGGGGCAAAGTGTTGACACTTTGGTCTTCATTCTTCTTGAGTTTCATGTGTTTCACAAATTGTATCTTATATCTTGGGTATT    BDBDCD@@E
C>;CCDEFFFFDE;AC>@71HCCGCG@=ECFEIHFCIGHFFBGHEIIG@IIGGEIJIIIHIIIJIJJHJGGJIGIIGIGF?DHHEBDDD@@B    RG:Z:0  NH:i:1  N
M:i:0   XT:A:U  md:Z:101
HWI-ST985:73:C08BWACXX:8:2103:17437:175854      99      chr1    3197333 254     66M6121N35M     =       3197379 6
268     TGAAGTGTCTGTTGGATTAATTAACTGCAATTCATCTCCAGTAAAATTTGGTAAGTTCCAATGTTTATGAAAGAAGAGTGGAGGATCCTGTTGGATTGTTT    @
CCFFDFFHHHHHJJJJJJJJJJJJJJJJJJHHIIJJJJJJJJJHJJJJJJJJJJJJJIJIJIJJJJJHHJJJJGIJJJJFHICGIIGHEEFFFFFEEDDEEDCDC    RG:Z:0  N
H:i:1   NM:i:1  XT:A:U  md:Z:66>6121*35
HWI-ST985:73:C08BWACXX:8:2103:17437:175854      147     chr1    3197379 254     20M6121N81M     =       3197333 -
6268    TTTGGTAAGTTCCAATGTTTATGAAAGAAGAGTGGAGGATCCTGTTGGATTGTTTGGCTGGACACTATTACATTGGAACTGTGTTCACAGAATCAAAGCTG    <
DDDDDEEECACFFFFFFHHHHHHHJJJJJJJJJJJJJJJJJJJJJJJJJJIHGJJJIJJJJJJIGHJJJJJIJJIJJJJJJJJJJJJJIJHHHHHFFFFFCCC    RG:Z:0  N
H:i:1   NM:i:1  XT:A:U  md:Z:81>6121*20
```

Cecilia Coimbra Klein

SAM format

Sequence **A**lignment/**M**ap



```
HWI-ST985:73:C08BWACXX:8:2208:2017:40383        147     chr1    3055454 180     101M    =       3055370 -185    T
TTGTTCCCAATACTAAGAAGGGGCAAAGTGTTGACACTTTGGTCTTCATTCTTCTTGAGTTTCATGTGTTTCACAAATTGTATCTTATATCTTGGGTATT     BDBDCD@@E
C>;CCDEFFFFDE;AC>@71HCCGCG@=ECFEIHFCIGHFFBGHEIIG@IIGGEIJIIIHIIIJIJJHJGGJIGIIGIGF?DHHEBDDD@@B    RG:Z:0  NH:i:1  N
M:i:0   XT:A:U  md:Z:101
HWI-ST985:73:C08BWACXX:8:2103:17437:175854      99      chr1    3197333 254     66M6121N35M     =       3197379 6
268     TGAAGTGTCTGTTGGATTAATTAACTGCAATTCATCTCCAGTAAAATTTGGTAAGTTCCAATGTTTATGAAAGAAGAGTGGAGGATCCTGTTGGATTGTTT    @
CCFFDFFHHHHHJJJJJJJJJJJJJJJJJJHHIIJJJJJJJJJHJJJJJJJJJJJJIJIJIJJJJJJHHJJJJGIJJJJFHICGIIGHEEFFFFFEEDDEEDCDC     RG:Z:0  N
H:i:1   NM:i:1  XT:A:U  md:Z:66>6121*35
```

Flag:
https://broadinstitute.github.io/picard/explain-flags.html

CIGAR:

- N → intron
- M → match
- I → insertion
- D → deletion
- S → soft-clip

More specification on SAM format:
https://samtools.github.io/hts-specs/SAMv1.pdf

Cecilia Coimbra Klein

BAM format

compressed binary representation of the SAM format

- specific block compression
  - BGZF
- support random access through the **index**

  ➡ fast retrieval of alignments overlapping a specified region

! BAM file must be sorted by genomic position (chromosome name and leftmost coordinate) in order to be indexed!

CRAM format

improved compressed binary representation of SAM

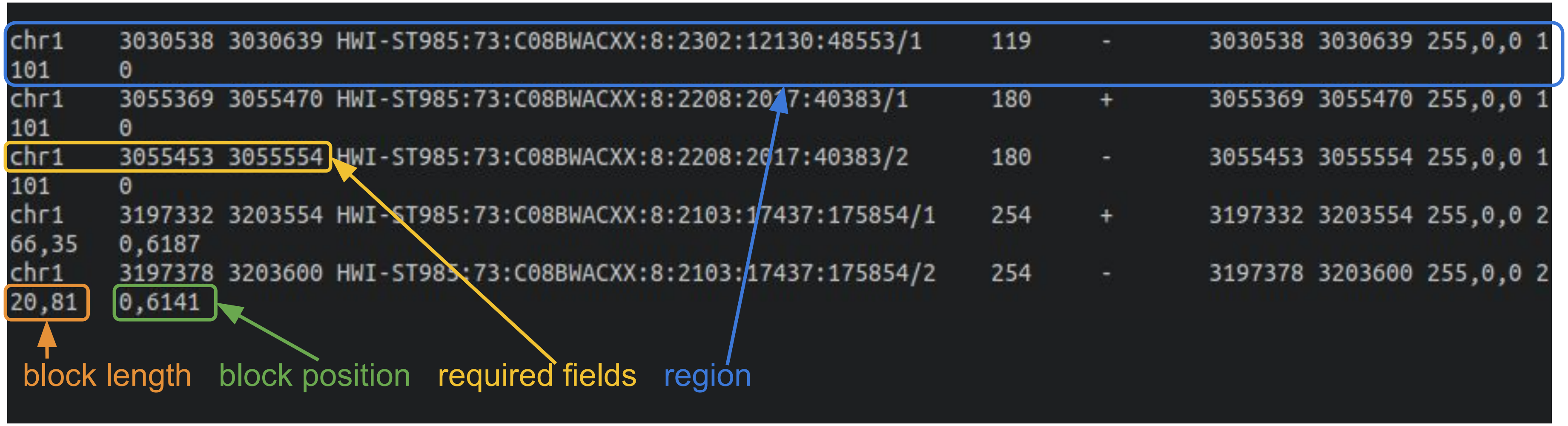
- different compression formats
  - gzip, bzip2, CRAM records
- CRAM records use different encoding strategies, e.g. bases are reference compressed by encoding base differences rather than storing the bases themselves
- random access support through the format itself (slices)

! CRAM indexing is external to the file format itself and may change independently of the file format specification in the future

BED format

provides a flexible and compact way to represent genomic regions (with breaks)

- 3 required fields + additional 9 fields
- more compact than GFF ➡ **tradeoff between size and provided information**



```
chr1    3030538 3030639 HWI-ST985:73:C08BWACXX:8:2302:12130:48553/1     119     -       3030538 3030639 255,0,0 1
101     0
chr1    3055369 3055470 HWI-ST985:73:C08BWACXX:8:2208:2017:40383/1      180     +       3055369 3055470 255,0,0 1
101     0
chr1    3055453 3055554 HWI-ST985:73:C08BWACXX:8:2208:2017:40383/2      180     -       3055453 3055554 255,0,0 1
101     0
chr1    3197332 3203554 HWI-ST985:73:C08BWACXX:8:2103:17437:175854/1    254     +       3197332 3203554 255,0,0 2
66,35   0,6187
chr1    3197378 3203600 HWI-ST985:73:C08BWACXX:8:2103:17437:175854/2    254     -       3197378 3203600 255,0,0 2
20,81   0,6141
```

10) blockCount - The number of blocks (exons) in the BED line.

11) blockSizes - A comma-separated list of the block sizes. The number of items in this list should correspond to *blockCount*.

12) blockStarts - A comma-separated list of block starts. All of the *blockStart* positions should be calculated relative to *chromStart*. The number of items in this list should correspond to *blockCount*.

https://genome.ucsc.edu/FAQ/FAQformat.html#format1

bedGraph and wig formats

bedGraph

- allows the display of continuous-valued data
- useful for probability scores and transcriptome data (CHIp-seq, RNA-seq)
- is a text file

```
track type=bedGraph name="BedGraph Format" description="BedGraph format" visibility=full color=200,100,0 altColor=0,100,200
priority=20
chr19 49302000 49302300 -1.0
chr19 49302300 49302600 -0.75
```

wig

- allows the display of continuous-valued data
- more compressed than bedGraph
- is a text file

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

bigBed, bigWig

Useful formats to display data on the UCSC genome browser

- BED, bedGraph, wig - are tab delimited text files
- bigBed, bigWig - are binary version of this files
- for each type of file there is a specific procedure to make a binary form
  - easily transferable
  - not so big
  - allows indexed access

Hands-on

Common file formats 2.4

https://public-docs.crg.es/rguigo/Data/cklein/courses/UVIC/handsOn/#_common_file_formats