TOPIC 3:

Sequence file formats and bioinformatic gotchas

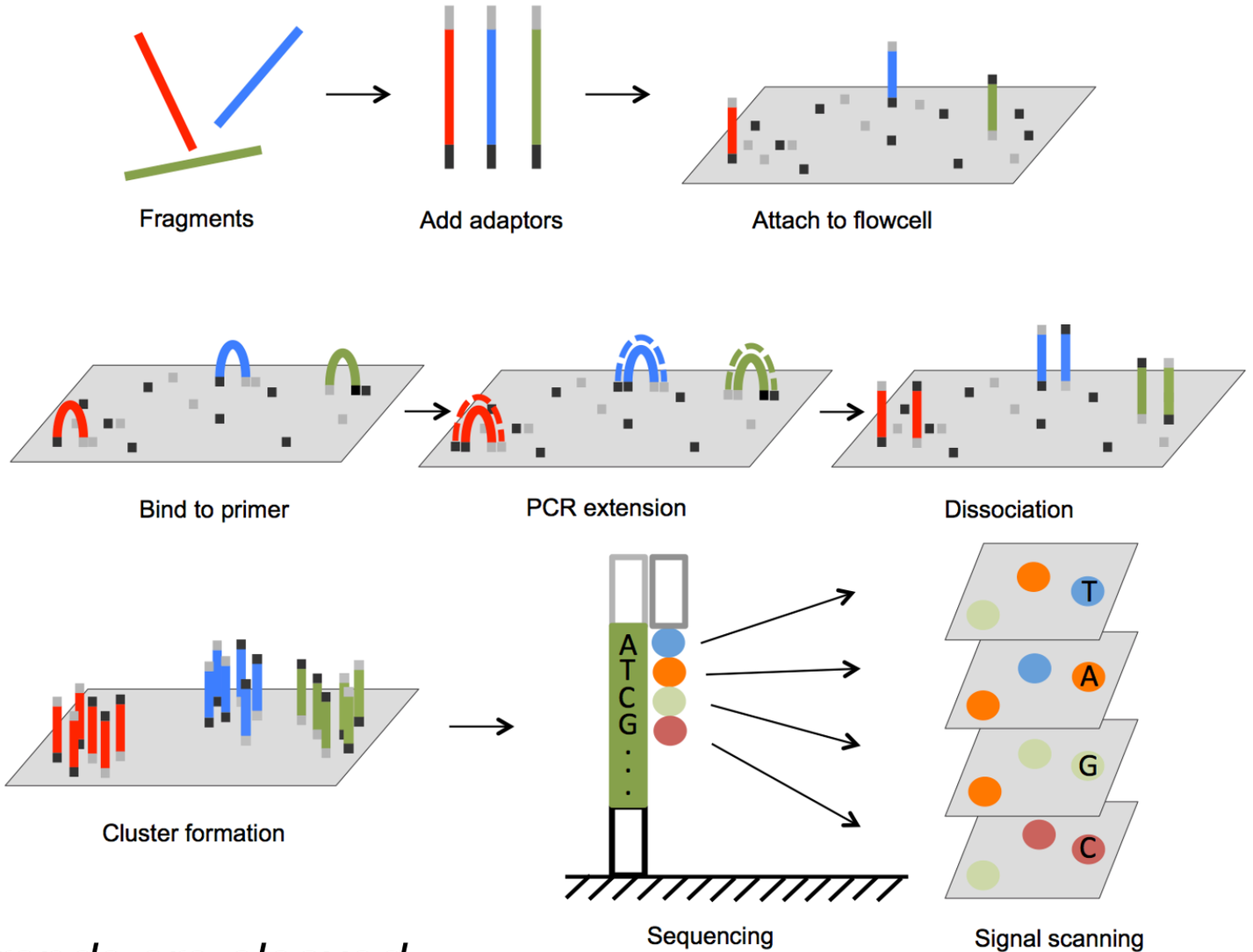
BIOL525D - Bioinformatics for Evolutionary Biology 2021

Outline

1. **Understand sequence file formats**
2. Preparing files for analysis
    - Tutorial looking at sequence data files
3. A tour through some bioinformatic gotchas
    - Short exercises

Part 1: Sequence file formats

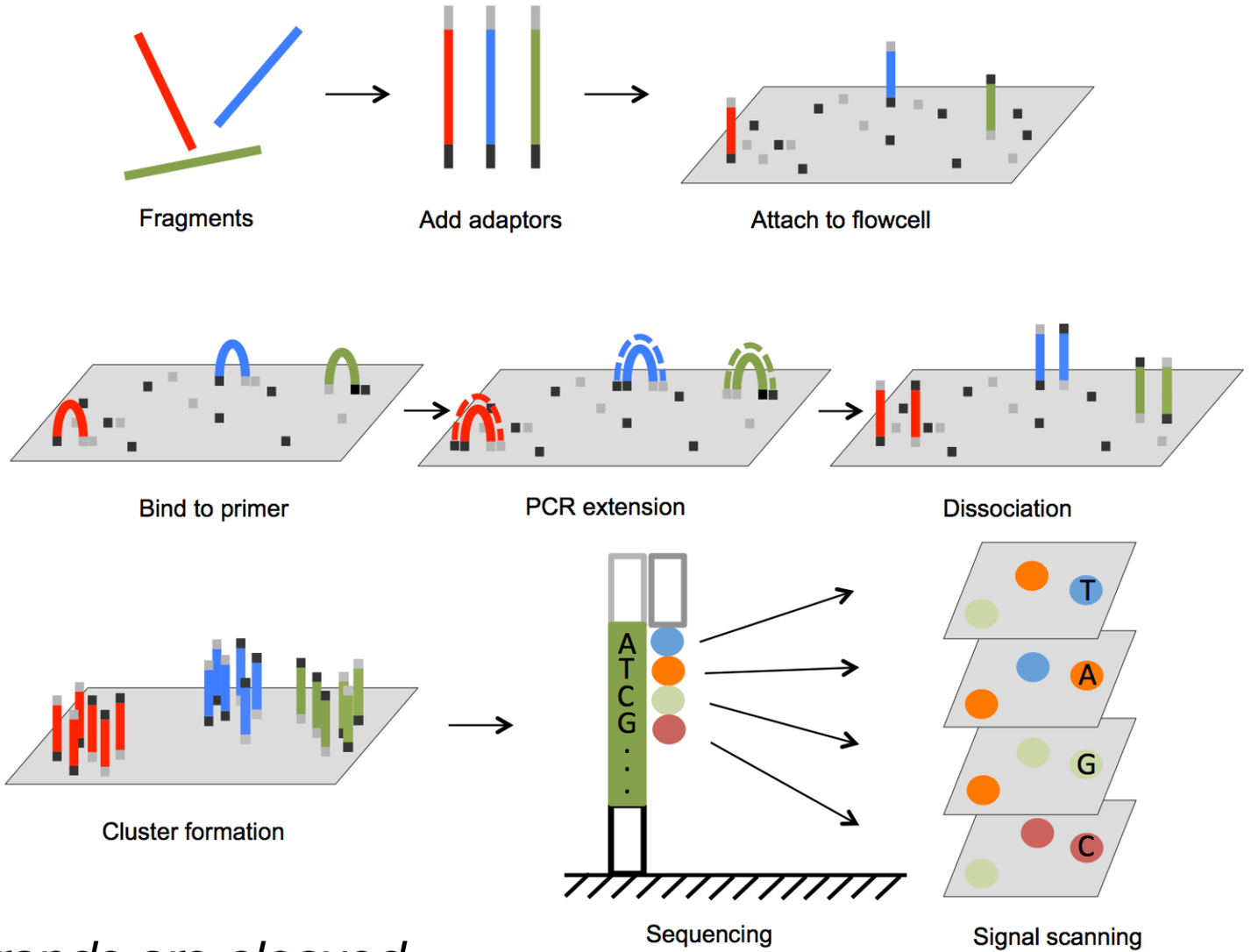
Illumina sequencing



Reverse strands are cleaved after cluster formation

Part 1: Sequence file formats

Illumina sequencing

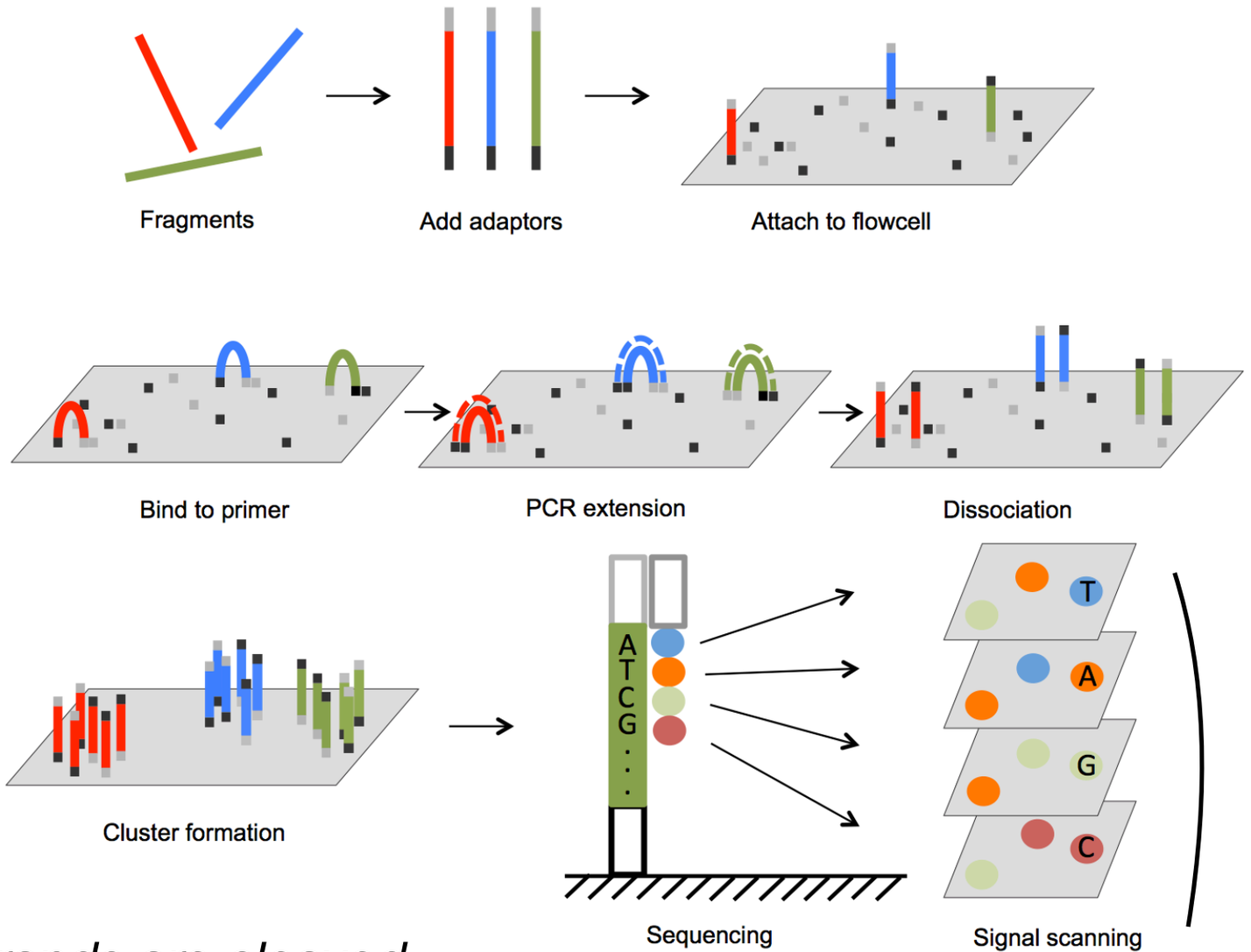


Reverse strands are cleaved after cluster formation

4 cycles are shown, but modern Illumina machines are capable of 600 cycles in one run

Part 1: Sequence file formats

Illumina sequencing



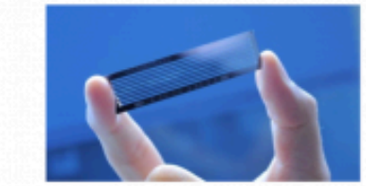
This process has generated 4 Images

Reverse strands are cleaved after cluster formation

4 cycles are shown, but modern Illumina machines are capable of 600 cycles in one run

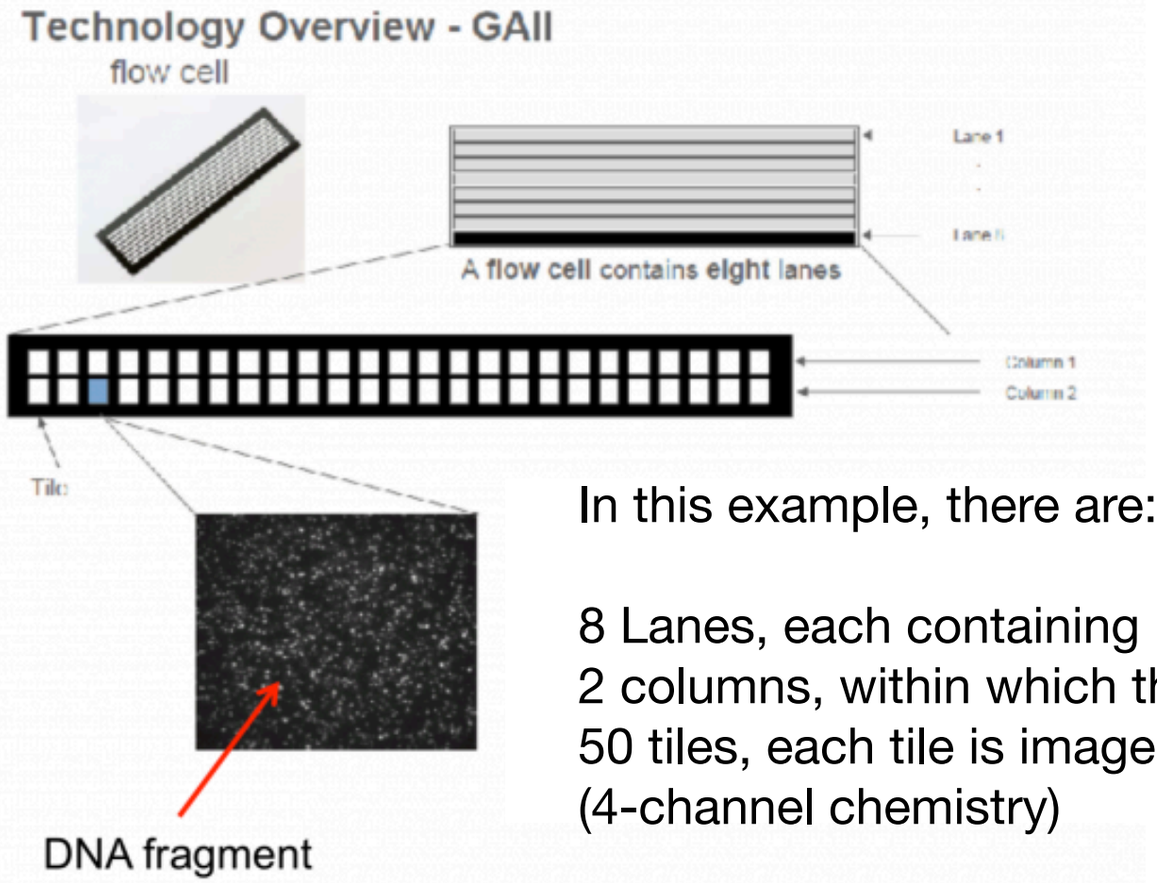
Part 1: Sequence file formats

Illumina sequencing



Chip, slide, flow cell…

HiSeq 2500

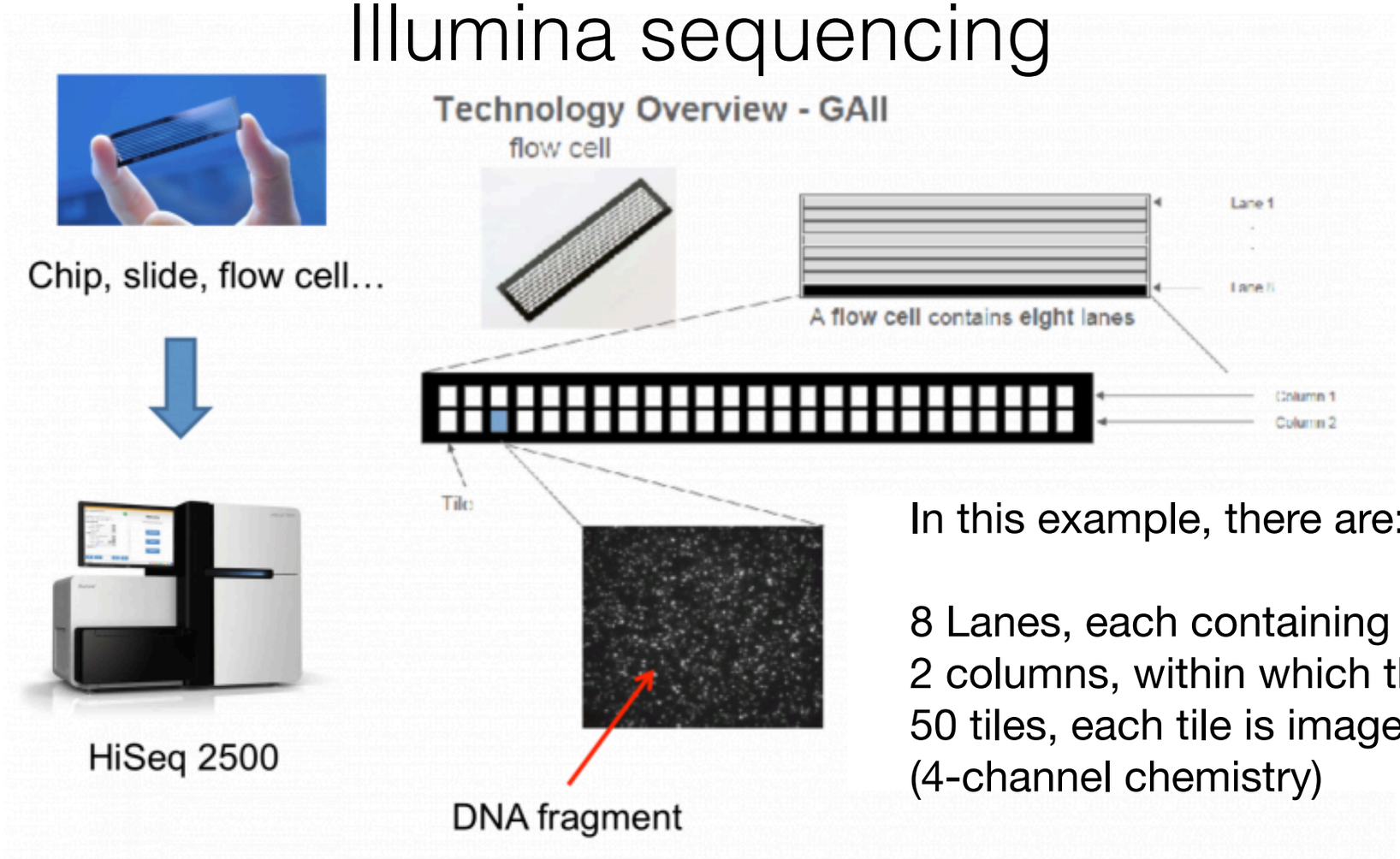


In this example, there are:

8 Lanes, each containing
2 columns, within which there are
50 tiles, each tile is imaged 4 times/cycle
(4-channel chemistry)

Part 1: Sequence file formats

Illumina sequencing



In this example, there are:

8 Lanes, each containing
2 columns, within which there are
50 tiles, each tile is imaged 4 times/cycle
(4-channel chemistry)

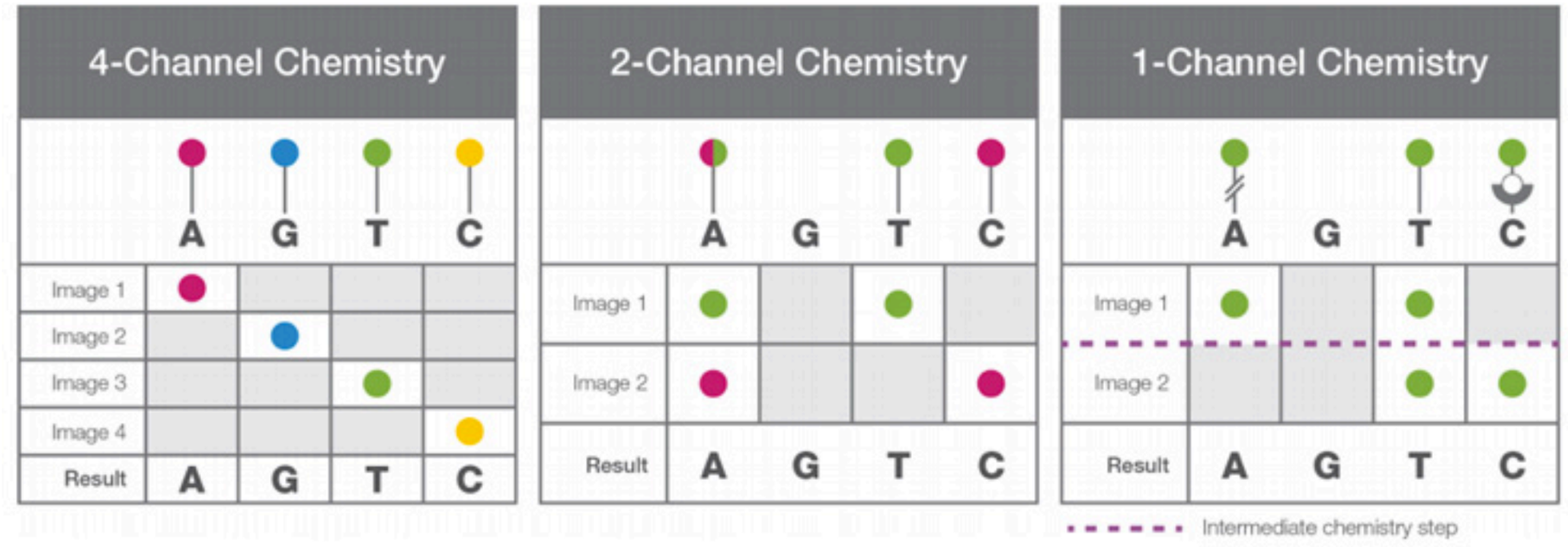
So there are approximately 8x2x50x4 = 3,000 images generated per cycle

Each image is about 3Mb in size

For an Illumina run using 300 cycles, that would be 3000x3x300 = 2,700,000 Mb of data (~2.7 Tb)

Part 1: Sequence file formats

Illumina sequencing

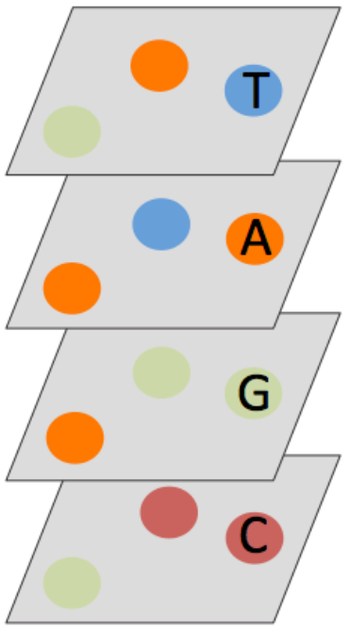


The number of channels refers to the numbers of colours the images detect

4-channel was Illumina's standard chemistry, but now 2-channel is more common

Part 1: Sequence file formats

Illumina sequencing



Using the stack of images from an Illumina machine you do the following:

1. Evaluates the light signal from every cluster to calculate the Quality Predictor Value (QPV), measuring things like:

The signal-to-noise ratio
Light Intensity

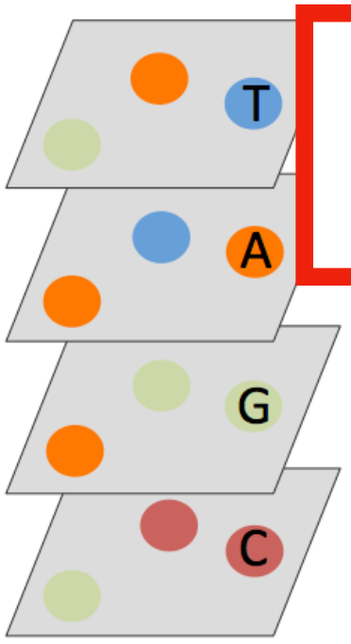
2. QPVs are converted into Phred quality scores (Q-scores) using a calibration curve built using previously sequenced samples

3. Convert the base call and the quality score into a FASTQ file

Part 1: Sequence file formats

Illumina sequencing

Using the stack of images from an Illumina machine you do the following:



1. Evaluates the light signal from every cluster to
suring
-noise ratio
Light Intensity

You'll probably never do these steps yourself, but it's good to know where the data come from!

2. QPVs are converted into Phred quality scores (Q-scores) using a calibration curve built using previously sequenced samples

3. Convert the base call and the quality score into a FASTQ file

Part 1: Sequence file formats

Remember this from yesterday?

What a FASTA file looks like:

Sequence name

```
>chr_1
TGGGCAAGGCTGATGAACAGCAGCTGCATAAATTCTCCCCTAATTATATTGTAAATAGCT
GCAGCACAACAATAAAGCTTTGTTAGAGACATCTAGAGAATCACACACTGCATCTGTTCT
GCCGCTCTCCCTCTTGCTCTGTTCTGAGAAGCACTTGTTCACTGATTCTGGGTTTGTATT
TGTGTTTTTCATGCTTAACATTGTTATTTGTTTGCCTAGAAAGTTCTTTGATTGGGCCAA
ATTAGTCGATTTTAAAGAGTGCACTTCTCTAGTGCATGTAATCTATGTGGACATCTCAAT
AGCTGCTTAATTTGTTTAGTGGTAATCTCCTCTGAACAGAGAGAAAGGCCTACATGCAGC
CCTCAGAGGAGAGGTGTCAATCTCTCTTTGATTATCTCTTTGTTTCCCTTCAGAAGAATC
ATTCTAATCTGGTATTGTACAAGAGGAAATAAATGGGACTAAAACCAGGCATGCACCATC
TGATAGATTCACATCCCTAGAAGACTTTTGTTGTGTTTGTTTCAAGTGGAGAGCCTGCTG
```

Nucleotide sequence

FASTAs are plain text files

Part 1: Sequence file formats

Anatomy of a FASTQ file:

4 Lines instead of 2

```
@SRR357068.1 D042KACXX:3:1101:2690:2160 length=101
NCATCGTCCGGTATGTAGAACAGGGGAACCGGACGTTTTCCAAGGCGTAGC
CATGTTAGACAAGGCGCAGATATAGGTGA
+SRR357068.1 D042KACXX:3:1101:2690:2160 length=101
#4=DBDDDHFHFFHIGHIIIJJJJJJJJJJJJJBHDAGHJGGGHIJHFFFFD
DEDCCDCCCCDDDDDBDBD>CDEE>C@CD
```

FASTQs are plain text files

Part 1: Sequence file formats

Anatomy of a FASTQ file:

1. Sequence ID
(begins with “@“ not “>”)

Typically contains information on the origin of the read - like which lane and tile it came from, where in the tile the cluster was located

```
@SRR357068.1 D042KACXX:3:1101:2690:2160 length=101
NCATCGTCCGGTATGTAGAACAGGGGAACCGGACGTTTTCCAAGGCGTAGC
CATGTTAGACAAGGCGCAGATATAGGTGA
+SRR357068.1 D042KACXX:3:1101:2690:2160 length=101
#4=DBDDDHFHFFHIGHIIIJJJJJJJJJJJJBHDAGHJGGGHIJHFFFFD
DEDCCDCCCCDDDDDBDBD>CDEE>C@CD
```

FASTQs are plain text files

Part 1: Sequence file formats

Anatomy of a FASTQ file:

1. Sequence ID (begins with “@“ not “>”)

2. Nucleotide sequence

```
@SRR357068.1 D042KACXX:3:1101:2690:2160 length=101
NCATCGTCCGGTATGTAGAACAGGGGAACCGGACGTTTTCCAAGGCGTAGC
CATGTTAGACAAGGCGCAGATATAGGTGA
+SRR357068.1 D042KACXX:3:1101:2690:2160 length=101
#4=DBDDDHFHFFHIGHIIIJJJJJJJJJJJJBHDAGHJGGGHIJHFFFFD
DEDCCDCCCCDDDDDBDBD>CDEE>C@CD
```

FASTQs are plain text files

Part 1: Sequence file formats

Anatomy of a FASTQ file:

1. Sequence ID (begins with “@“ not “>”)

2. Nucleotide sequence

3. Spacer (always a “+”) with optional Sequence ID

```
@SRR357068.1 D042KACXX:3:1101:2690:2160 length=101
NCATCGTCCGGTATGTAGAACAGGGGAACCGGACGTTTTCCAAGGCGTAGC
CATGTTAGACAAGGCGCAGATATAGGTGA
+SRR357068.1 D042KACXX:3:1101:2690:2160 length=101
#4=DBDDDHFHFFHIGHIIIJJJJJJJJJJJJJBHDAGHJGGGHIJHFFFFD
DEDCCDCCCCDDDDDBDBD>CDEE>C@CD
```

FASTQs are plain text files

Part 1: Sequence file formats

Anatomy of a FASTQ file:

1. Sequence ID (begins with “@“ not “>”)

2. Nucleotide sequence

```
@SRR357068.1 D042KACXX:3:1101:2690:2160 length=101
NCATCGTCCGGTATGTAGAACAGGGGAACCGGACGTTTTCCAAGGCGTAGC
CATGTTAGACAAGGCGCAGATATAGGTGA
+SRR357068.1 D042KACXX:3:1101:2690:2160 length=101
#4=DBDDDHFHFFHIGHIIIJJJJJJJJJJJJJBHDAGHJGGGHIJHFFFFD
DEDCCDCCCCDDDDDBDBD>CDEE>C@CD
```

3. Spacer (always a “+”) with optional Sequence ID

4 Base quality scores (Q-scores)

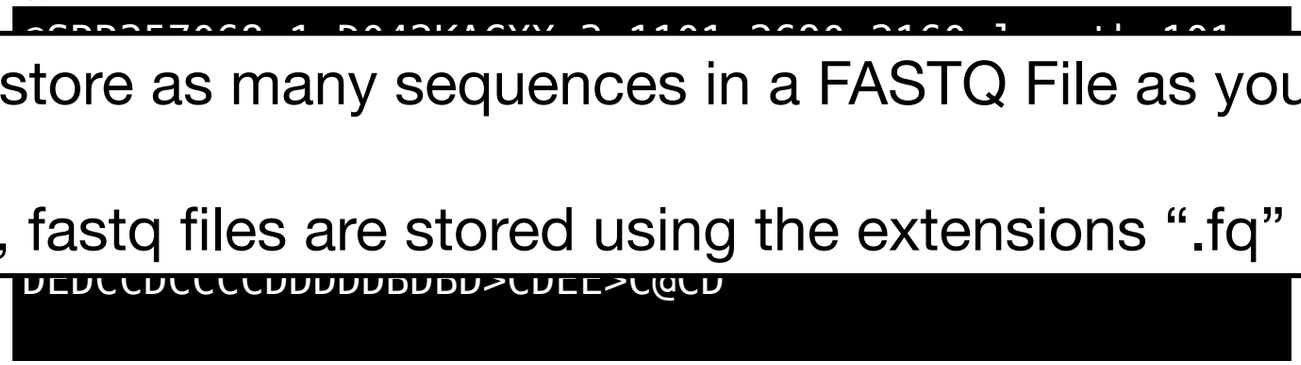
FASTQs are plain text files

Part 1: Sequence file formats

Anatomy of a FASTQ file:

1. Sequence ID (begins with “@“ not “>”)

2. Nucleotide sequence



You can store as many sequences in a FASTQ File as you like

By convention, fastq files are stored using the extensions “.fq” or “.fastq”

3. Spacer (always a “+”) with optional Sequence ID

4 Base quality scores (Q-scores)

FASTQs are plain text files

Part 1: Sequence file formats

Base quality scores (Q-scores)

$$Q_{Sanger} = -10log_{10}(p)$$

Where *p* is the probability that a base call is incorrect

$$Q_{Solexa} = -10log_{10}(\frac{p}{1-p})$$

Remember, those probabilities are calculated using the QPVs in Illumina sequencing

Part 1: Sequence file formats

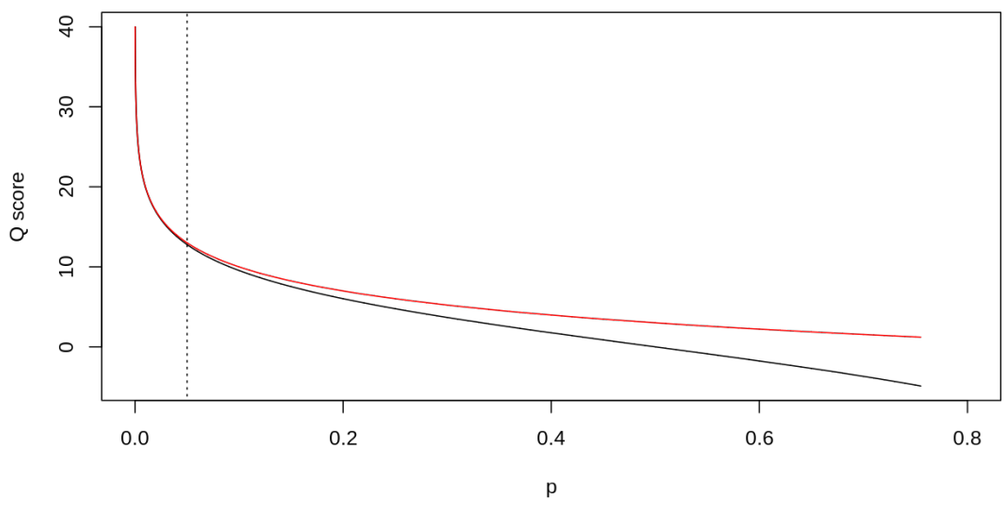
Base quality scores (Q-scores)

$$Q_{Sanger} = -10log_{10}(p)$$

Where *p* is the probability that a base call is incorrect

$$Q_{Solexa} = -10log_{10}(\frac{p}{1-p})$$

Remember, those probabilities are calculated using the QPVs in Illumina sequencing



***Red* line is Sanger**
Black line is Solexa

Asympotically identical when p is small

Part 1: Sequence file formats

Base quality scores (Q-scores)

$$Q_{Sanger} = -10log_{10}(p)$$

What's the probability that the base is incorrect if Q=30?

Part 1: Sequence file formats

Base quality scores (Q-scores)

$$Q_{Sanger} = -10log_{10}(p)$$

What's the probability that the base is incorrect if Q=30?

p[Q30] = 0.001

p[Q20] = 0.01

p[Q10] = 0.1

Part 1: Sequence file formats

Base quality scores (Q-scores)

You probably noticed that the Q-scores in the FastQ files are not numeric

```
@SRR357068.1 D042KACXX:3:1101:2690:2160 length=101
NCATCGTCCGGTATGTAGAACAGGGGAACCGGACGTTTTCCAAGGCGTAGC
CATGTTAGACAAGGCGCAGATATAGGTGA
+SRR357068.1 D042KACXX:3:1101:2690:2160 length=101
#4=DBDDDHFHFFHIGHIIIJJJJJJJJJJJJJBHDAGHJGGGHIJHFFFFD
DEDCCDCCCCDDDDDBDBD>CDEE>C@CD
```

Under Illumina sequencing,
ASCII encoding is used to refer to Q scores from 0 to 62

Slightly different encoding strategies are used by the different technologies

Part 1: Sequence file formats

PacBio Sequencing

Nanopore Sequencing

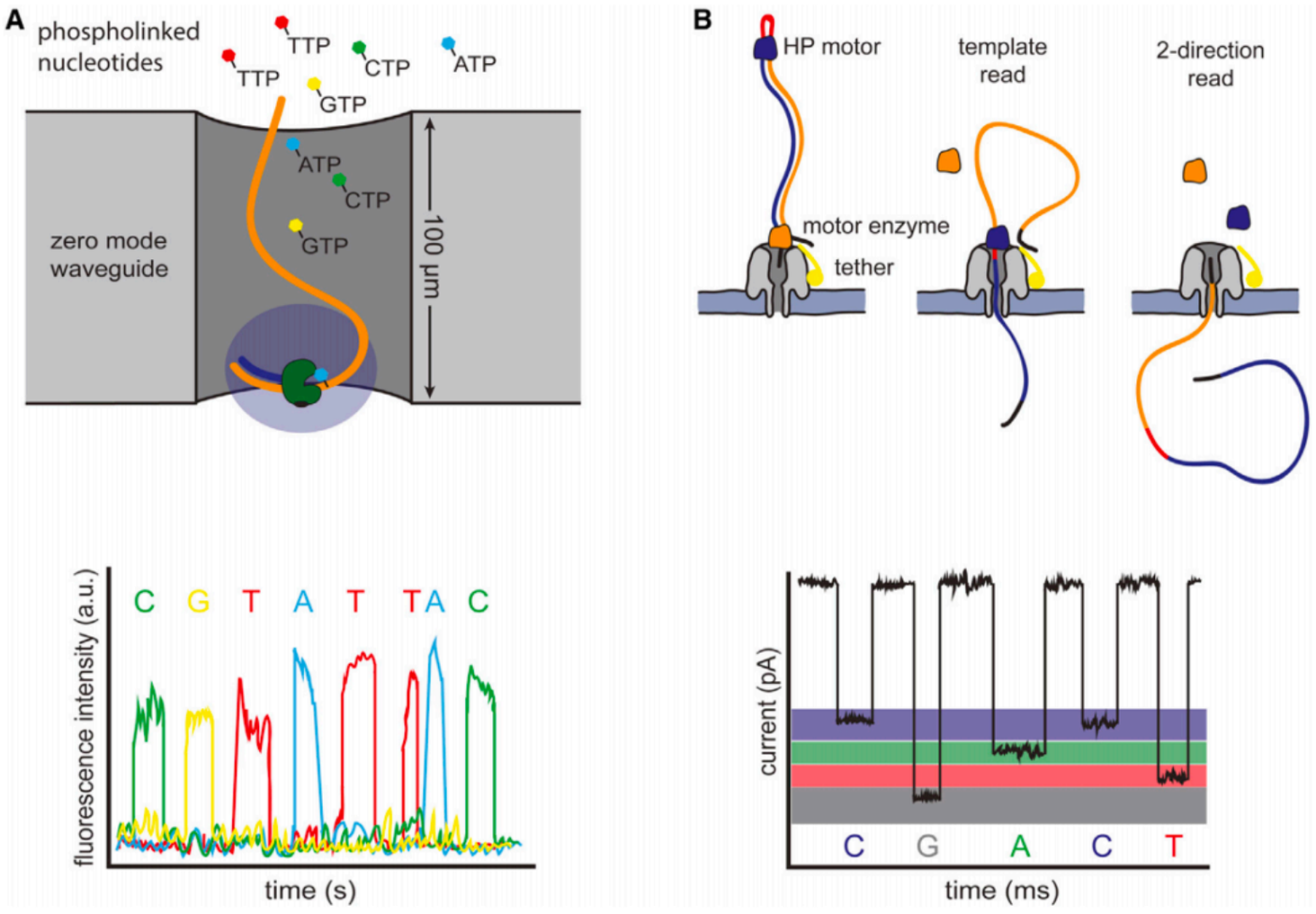


Figure 3. Single Molecule Sequencing Platforms

(A) Pacific Bioscience's SMRT sequencing. A single polymerase is positioned at the bottom of a ZMW. Phosphate-labeled versions of all four nucleotides a present, allowing continuous polymerization of a DNA template. Base incorporation increases the residence time of the nucleotide in the ZMW, resulting i detectable fluorescent signal that is captured in a video.

(B) Oxford Nanopore's sequencing strategy. DNA templates are ligated with two adapters. The first adaptor is bound with a motor enzyme as well as a teth whereas the second adaptor is a hairpin oligo that is bound by the HP motor protein. Changes in current that are induced as the nucleotides pass through the p are used to discriminate bases. The library design allows sequencing of both strands of DNA from a single molecule (two-direction reads).

Excerpted from Reuter et al 2015 - Molecular Cell

Outline

1. Understand sequence file formats
2. **Preparing files for analysis**
    - Tutorial looking at sequence data files
3. A tour through some bioinformatic gotchas
    - Short exercises

Part 2: Preparing files for analysis

What do you do when you get your data?

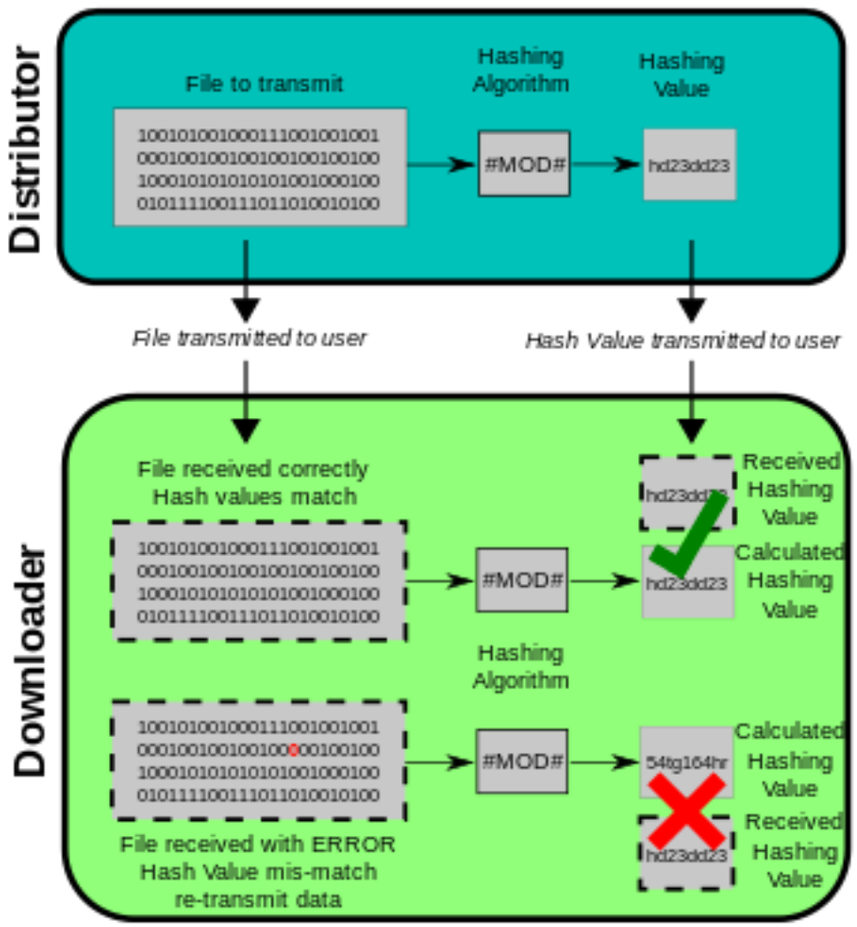
Part 2: Preparing files for analysis

1) Check files for completeness, use checksums if file corruption is suspected

Downloading large data files takes a long time

There is a possibility of data corruption when files are downloaded

There are command line tools for verifying data complete-ness



MD5 and SHA-1 are the most common checksum methods

There is a short demonstration using SHA-1 sums in the tutorial

Part 2: Preparing files for analysis

1) Check files for completeness, use checksums if file corruption is suspected

2) Inspect quality statistics

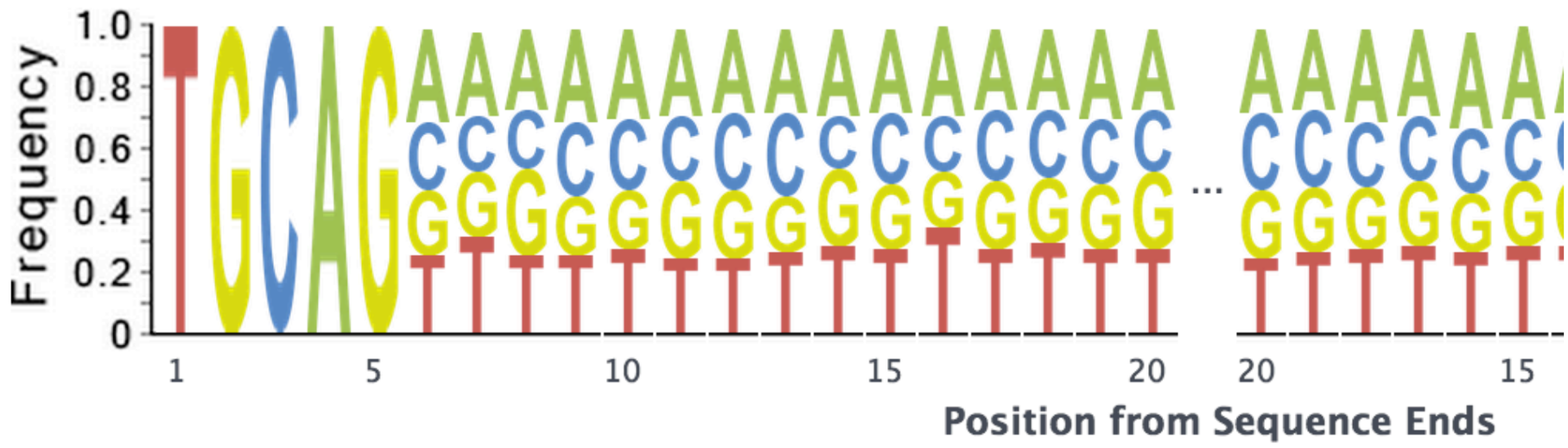
There are many possible statistics to query:

- Number and length of sequences
- Base qualities**
- Poly A/T tails
- Presence of tag sequences (things that you added during library prep.)
- Sequence complexity (e.g. identify repetitive data ATATATATATATATATATATATATA)

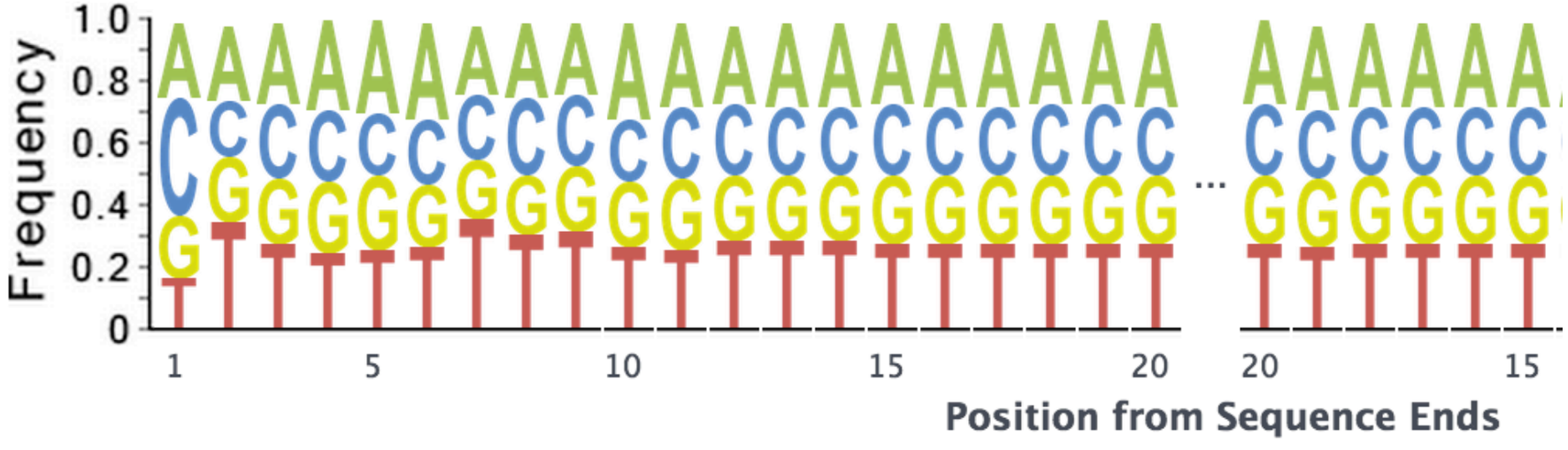
There are standard tools for examining these, such as prinseq and fastqc

Part 2: Preparing files for analysis

Distribution of base frequencies in GBS reads - with enzyme cut site

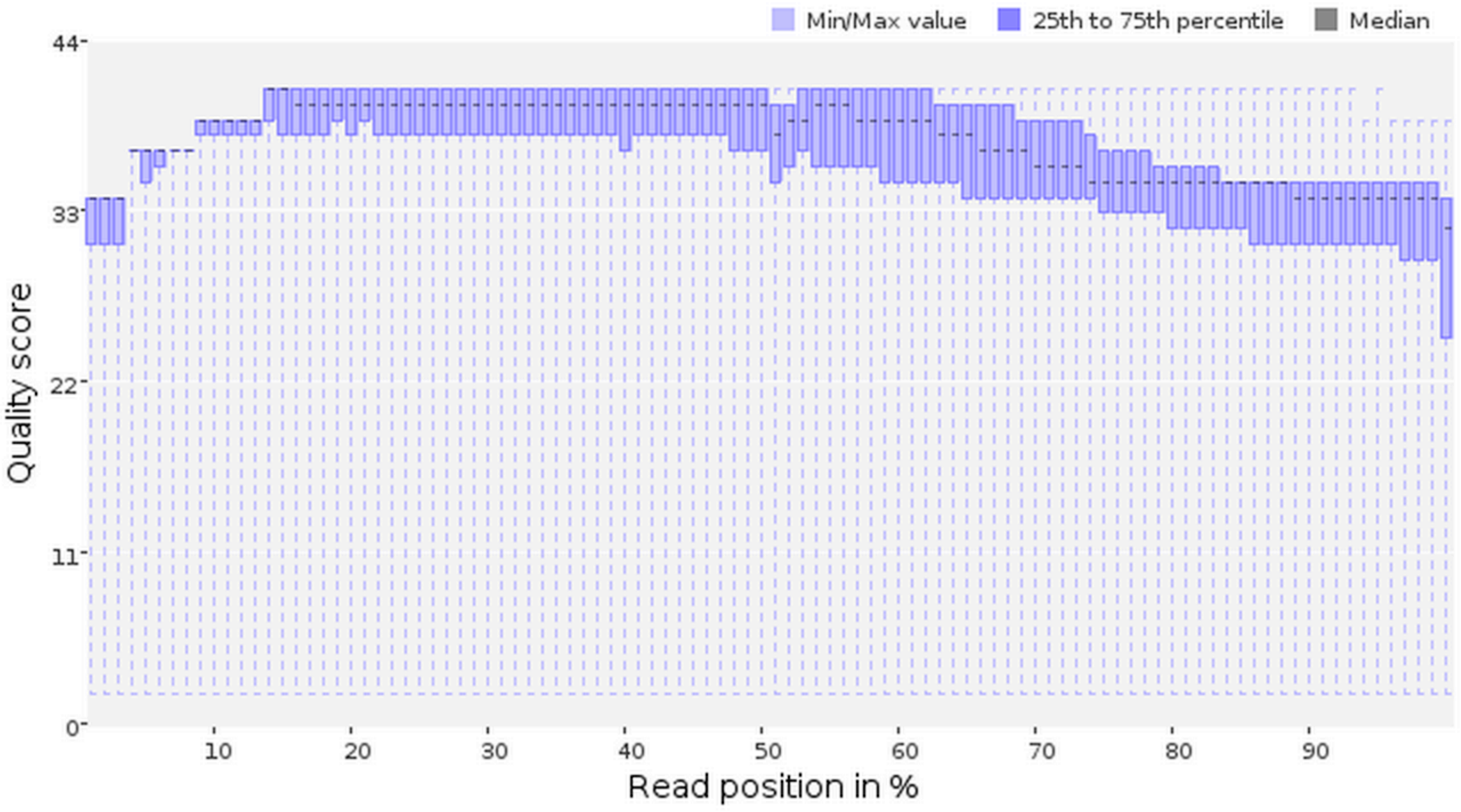


Distribution in RNAseq data - with no adapters/tags used



Part 2: Preparing files for analysis

A typical quality score distribution for Illumina reads



Part 2: Preparing files for analysis

1) Check files for completeness, use checksums if file corruption is suspected

2) Inspect quality statistics

3) Possible steps to clean files

– De-multiplex

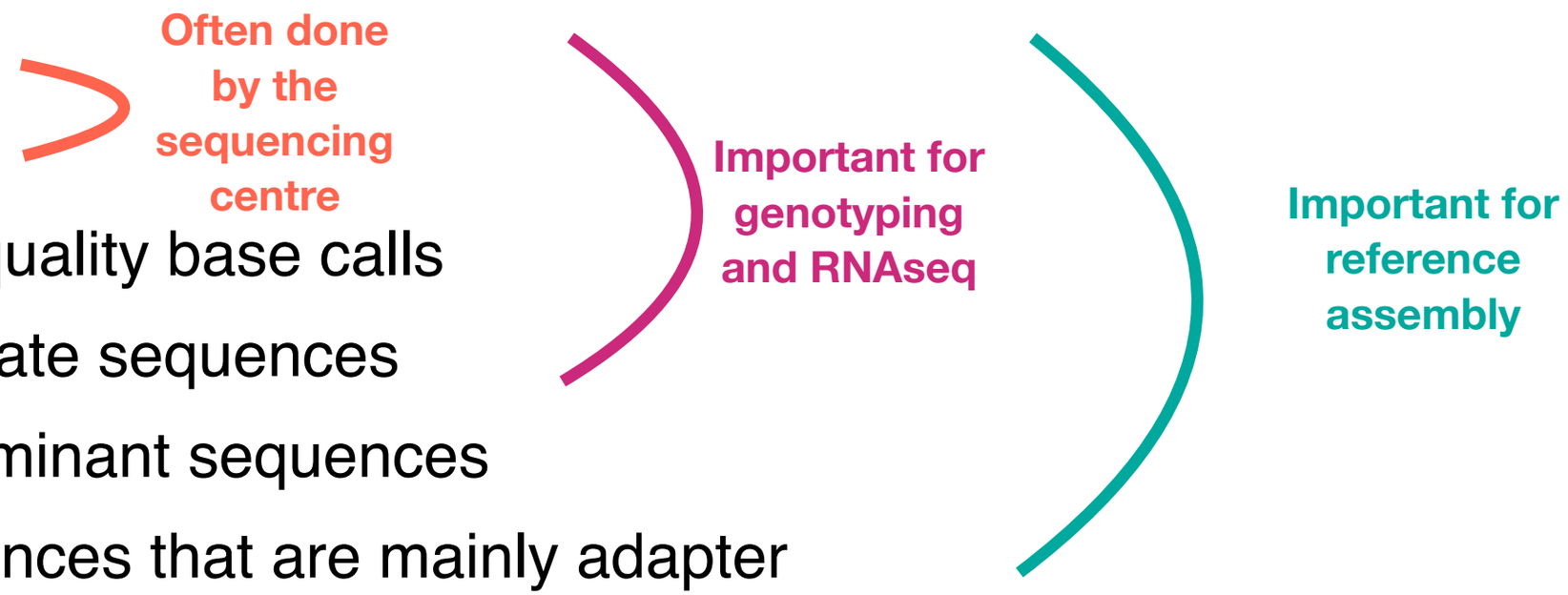
– Trim adapters

– Filter/trim low quality base calls

– Remove duplicate sequences

– Remove contaminant sequences

– Remove sequences that are mainly adapter



Many programs to implement these steps!

Part 2: Preparing files for analysis

Quality trimming

Choice of quality score to filter to depends upon the application:

- Too low a quality score cutoff:
    1. increase run times and RAM usage
    2. Bad results (e.g. false SNP calls)
- Too high a quality score cutoff:
    1. Faster run times
    2. Potentially lose useful data (e.g. more fragmented assemblies or missing SNPs)

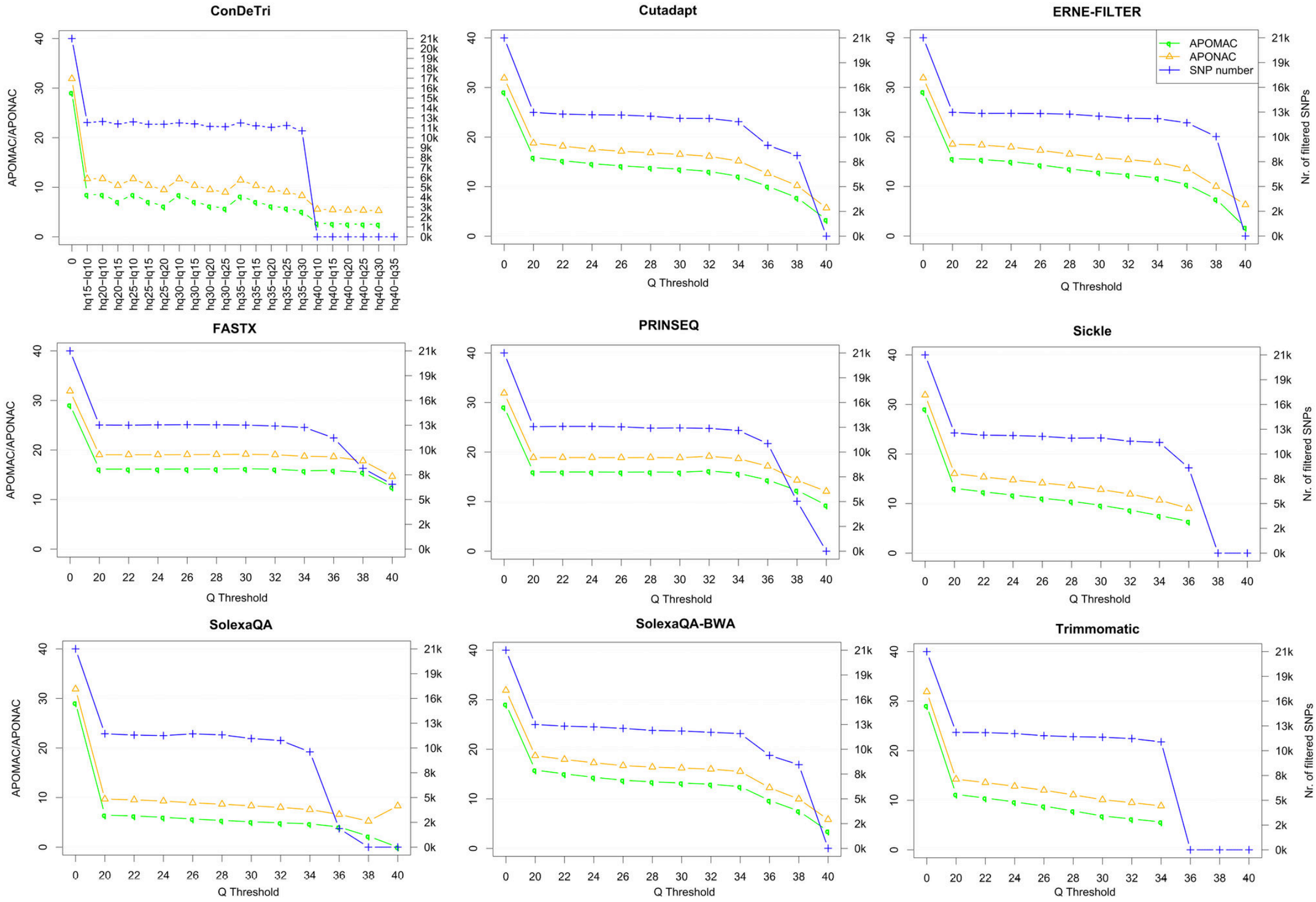
Q20 is a rule of thumb, but it depends on what you're doing

Part 2: Preparing files for analysis

Quality trimming

Del Fabbro et al 2013

Number of variants detected



Part 2: Preparing files for analysis

Duplicate reads

PCR is a common feature of many library preps

That can introduce errors and bias impacting downstream analysis

A large number of duplicates is potentially a sign of wasted sequencing effort

```
                                                *
TTTCATACTAACTAGCCTGCGGTCTGTGTTTCCCGACTTCTGAGTCATGGGGTTTCAATGCCTATAGATTC
              ........................C.
               .........................
                    ............T............
                     ................C.........
                         ........................
                            ........................
                               ........................
                                  .....C..................
                                     ..C.....................
```

```
                                                *
TTTCATACTAACTAGCCTGCGGTCTGTGTTTCCCGACTTCTGAGTCATGGGGTTTCAATGCCTATAGATTC
                          ........................
                               ............T............
                                ................C.........
                                ................C.........
                                ................C.........
                                ................C.........
                                        ........................
                                        ........................
                                        ........................
```

High numbers of duplicates may be expected in some cases (e.g. GBS, RNAseq)

A common strategy is to identify and tag reads that look like duplicates using PicardTools (now part of GATK)

Part 2: Preparing files for analysis

Contamination

Contamination in your samples can lead to big errors downstream

Latest

The Atlantic

SCIENCE

Inside the Bizarre Genome of the World's Toughest Animal

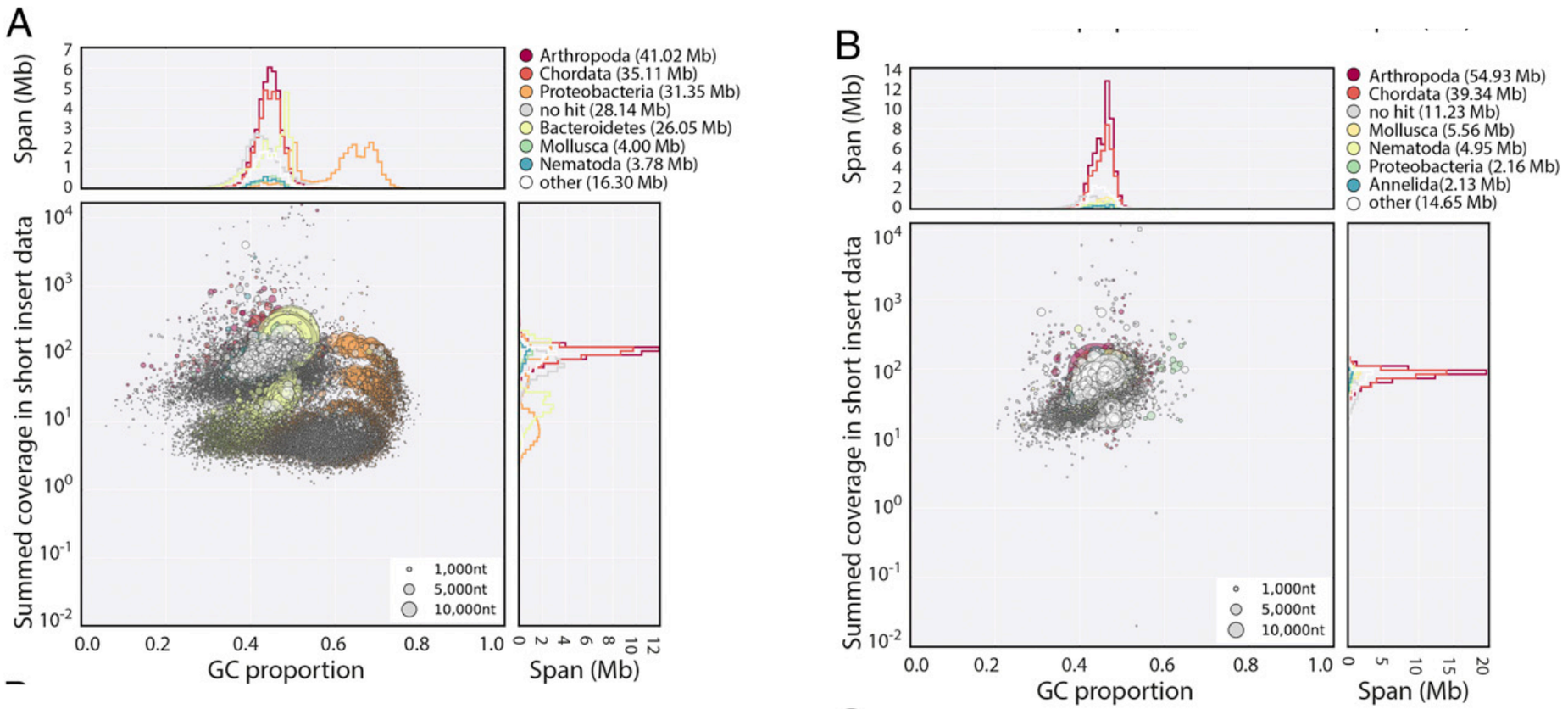
Tardigrades are sponges for foreign genes. Does that explain why they are famously indestructible?

Koutsovoulosa et al 2016 PNAS

Part 2: Preparing files for analysis

Contamination

Contamination in your samples can lead to big errors downstream

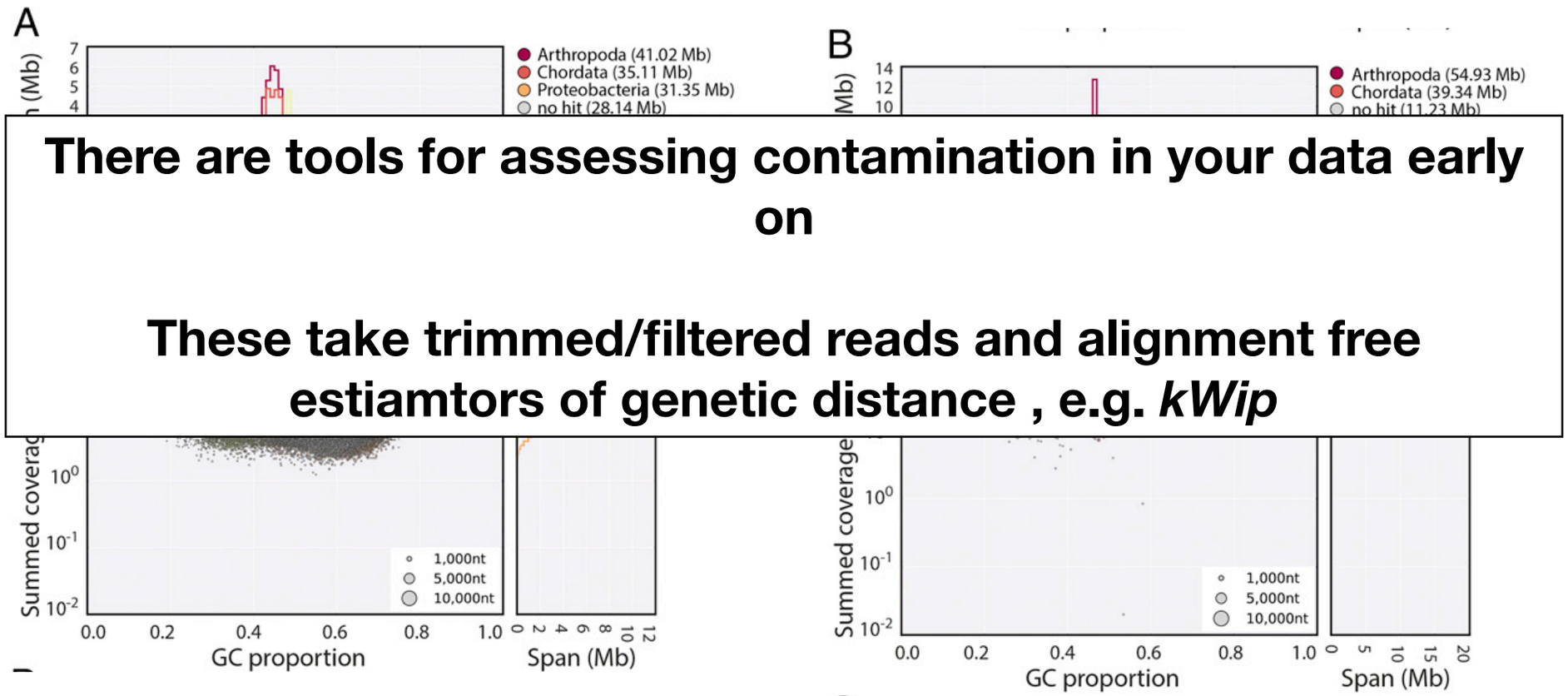


Koutsovoulosa et al 2016 PNAS

Part 2: Preparing files for analysis

Contamination

Contamination in your samples can lead to big errors downstream



There are tools for assessing contamination in your data early on

These take trimmed/filtered reads and alignment free estiamtors of genetic distance , e.g. *kWip*

Koutsovoulosa et al 2016 PNAS

Outline

1. Understand sequence file formats
2. Preparing files for analysis
    - **Tutorial looking at sequence data files**
3. A tour through some bioinformatic gotchas

- Short exercises

Tutorial:
Work through part 1 of the tutorial associated with this session

How many sequences are in the FASTQ files you are looking at?

How do you think programs identify read-pairs using FASTQ files?

Part 3: Bioinformatic gotchas

We asked evolutionary biologists to tell us the things that get tripped up on, or that they wish they'd known starting out

Part 3: Bioinformatic gotchas

Paired end reads

If one or other of a read pair is removed, you'll need to identify the mate and eliminate the orphan reads

Some programs output reads in paired and unpaired files (e.g. prinseq and Trimmomatic) others do not…

In such case, you won't have _R1.fastq and _R2.fastq files so have to repair the data by re-pairing your reads

Part 3: Bioinformatic gotchas

Genomic coordinates/annotations

The Salmon's genes are stored in a GFF file (stands for "General Feature Format")

```
chr_1   Gnomon  exon    71829   74369   .       -       .       ID=exon-XM_024393305.1-16;Parent=rna-XM_024393305.1;Dbxref=Gene
chr_1   Gnomon  gene    71829   332783  .       -       .       ID=gene-LOC112228197;Dbxref=GeneID:112228197;Name=LOC112228197;
chr_1   Gnomon  mRNA    71829   332783  .       -       .       ID=rna-XM_024393305.1;Parent=gene-LOC112228197;Dbxref=GeneID:11
chr_1   Gnomon  CDS     73661   74369   .       -       1       ID=cds-XP_024249073.1;Parent=rna-XM_024393305.1;Dbxref=GeneID:1
chr_1   Gnomon  CDS     74541   74650   .       -       0       ID=cds-XP_024249073.1;Parent=rna-XM_024393305.1;Dbxref=GeneID:1
chr_1   Gnomon  exon    74541   74650   .       -       .       ID=exon-XM_024393305.1-15;Parent=rna-XM_024393305.1;Dbxref=Gene
chr_1   Gnomon  CDS     82414   82521   .       -       0       ID=cds-XP_024249073.1;Parent=rna-XM_024393305.1;Dbxref=GeneID:1
chr_1   Gnomon  exon    82414   82521   .       -       .       ID=exon-XM_024393305.1-14;Parent=rna-XM_024393305.1;Dbxref=Gene
chr_1   Gnomon  CDS     86180   86329   .       -       0       ID=cds-XP_024249073.1;Parent=rna-XM_024393305.1;Dbxref=GeneID:1
chr_1   Gnomon  exon    86180   86329   .       -       .       ID=exon-XM_024393305.1-13;Parent=rna-XM_024393305.1;Dbxref=Gene
```

Sequence name

Source

Feature

Start

End

Score

Strand

Frame

Attributes

GFFs, GTFs and BEDs are plain text files

Part 3: Bioinformatic gotchas

Genomic coordinates/annotations

The Salmon's genes are stored in a GFF file (stands for "General Feature Format")

```
chr_1   Gnomon  exon    71829   74369   .       -       .       ID=exon-XM_024393305.1-16;Parent=rna-XM_024393305.1;Dbxref=Gene
chr_1   Gnomon  gene    71829   332783  .       -       .       ID=gene-LOC112228197;Dbxref=GeneID:112228197;Name=LOC112228197;
chr_1   Gnomon  mRNA    71829   332783  .       -       .       ID=rna-XM_024393305.1;Parent=gene-LOC112228197;Dbxref=GeneID:11
chr_1   Gnomon  CDS     73661   74369   .       -       1       ID=cds-XP_024249073.1;Parent=rna-XM_024393305.1;Dbxref=GeneID:1
chr_1   Gnomon  CDS     74541   74650   .       -       0       ID=cds-XP_024249073.1;Parent=rna-XM_024393305.1;Dbxref=GeneID:1
chr_1   Gnomon  exon    74541   74650   .       -       .       ID=exon-XM_024393305.1-15;Parent=rna-XM_024393305.1;Dbxref=Gene
chr_1   Gnomon  CDS     82414   82521   .       -       0       ID=cds-XP_024249073.1;Parent=rna-XM_024393305.1;Dbxref=GeneID:1
chr_1   Gnomon  exon    82414   82521   .       -       .       ID=exon-XM_024393305.1-14;Parent=rna-XM_024393305.1;Dbxref=Gene
chr_1   Gnomon  CDS     86180   86329   .       -       0       ID=cds-XP_024249073.1;Parent=rna-XM_024393305.1;Dbxref=GeneID:1
chr_1   Gnomon  exon    86180   86329   .       -       .       ID=exon-XM_024393305.1-13;Parent=rna-XM_024393305.1;Dbxref=Gene
```

Attributes

Depending on the version you're using, items in the attribute string may be separated by a space (" ") or an equals sign "="

GFFs, GTFs and BEDs are plain text files

Part 3: Bioinformatic gotchas

Genomic coordinates/annotations

It's very common to store lists of genomic coordinates

Minimal example of a BED file

```
chr_1   799999  1000000
chr_1   3099999 3100000
chr_1   4199999 4300000
chr_2   132316  202321
chr_2   1253129 1312314
chr_2   2012399 2112111
```

Sequence name

Start

End

BED files are extremely common

They share features with GFF/GTF files

GFFs, GTFs and BEDs are plain text files

Part 3: Bioinformatic gotchas

GTF Vs. BED Files

It's very common to store lists of genomic coordinates

```
chr_1   799999  1000000
chr_1   3099999 3100000
chr_1   4199999 4300000
chr_2   132316  202321
chr_2   1253129 1312314
chr_2   2012399 2112111
```

← The top entry refers to locations 800,000 to 1,000,000

This would be listed as:

chr1 source element 800000 1000000

In a GFF file

Part 3: Bioinformatic gotchas

0-Based vs. 1-Based Indexing

Python starts counting indices at 0
R starts at 1

0-Based

1-Based

For example, when accessing the first element in an array

```
>>> x = ["a", "b", "c"]
>>> x[1]
'b'
>>> x[0]
'a'
>>>
```

```
> x <- c("a","b","c")
> x[1]
[1] "a"
```

Part 3: Bioinformatic gotchas

Naming things

Given the flexibility of the command line, we can pretty much name our files and variables whatever we like as long as we can keep track

Obviously it's best to make things interpretable by someone else the following is probably not a great idea (I'm guilty of this one!)

```
$ ls
./
../
my file version 1.txt
myFile1.temp.txt
myFile1.txt
myFile1.v1.txt
myFile2.junk.txt
myFile2.junk.txt.2
```

You'll thank yourself for doing this right the first time!!

Additionally, some packages (particularly phylogenetic software) read metadata from the names of files, so they have to be named a certain way

Part 3: Bioinformatic gotchas

Naming things

Issues can arise when using white space and special characters!

```
s0784966@SCE-BIO-C03959 : ~/workingJunk
$ ls
./                          ../
my file version 1.txt  readSets1&2.txt
(base)
s0784966@SCE-BIO-C03959 : ~/workingJunk
$ more my files version 1.txt
my: No such file or directory
files: No such file or directory
version: No such file or directory
1.txt: No such file or directory
(base)
s0784966@SCE-BIO-C03959 : ~/workingJunk
$ more my\ file\ version\ 1.txt
Remember Tom, you are cool.
Love, Mum
(base)
s0784966@SCE-BIO-C03959 : ~/workingJunk
$ cat readSets1&2.txt
[1] 7863
-bash: 2.txt: command not found
cat: readSets1: No such file or directory
[1]+  Exit 1                  cat readSets1
(base)
s0784966@SCE-BIO-C03959 : ~/workingJunk
$ cat readSets1\&2.txt
Tom, you are handsome charming and witty(base)
```

Here I have two files:

my file version 1.txt
readSets_1&2.txt

The use of the white space and special “&” character mean we need to use the escape character (“\”)

This can be easily forgotten when moving from GUI to the command line

Part 3: Bioinformatic gotchas

Chromosomes or scaffolds

A lot of software was developed with analysis of humans in mind

Humans have 23 chromosome pairs - and a really good reference genome

Some analysis software has limits on the number of chromosome in a reference genome

Draft genome assemblies may have many, many scaffolds rather than fully assembled chromsomes

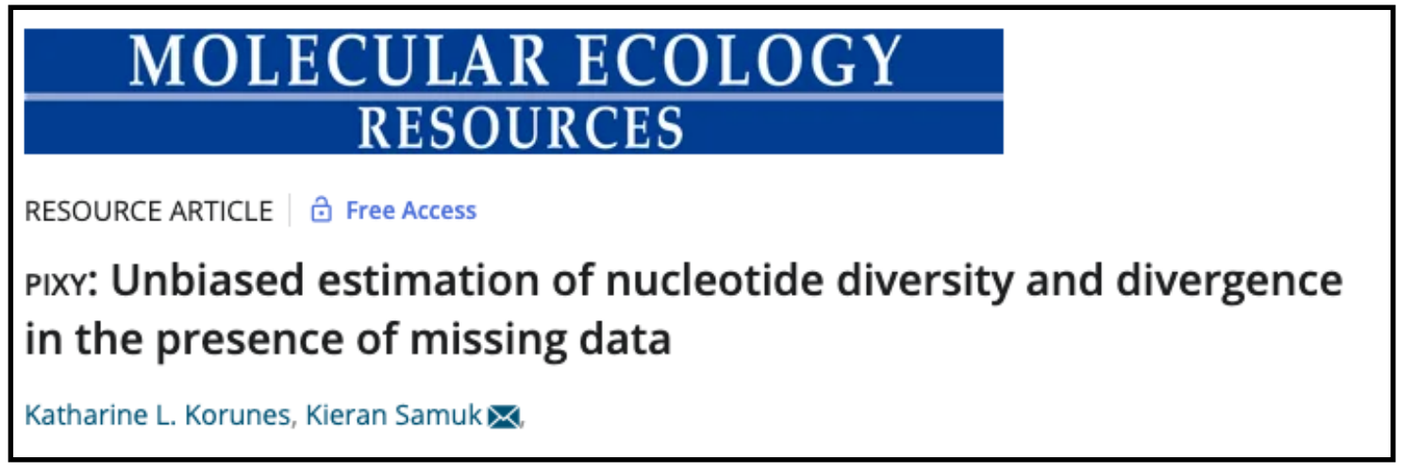
chr_1, chr1, chromosome_1, 1

Part 3: Bioinformatic gotchas

A lot of software was developed with analysis of humans in mind

In many evolutionary analysis, we are interested in the sites that do not exhibit variation as well as those with variants

However, many analysis software packages assume that invariant sites are homozygous for the reference allele, but that is a big assumption that can lead to problems



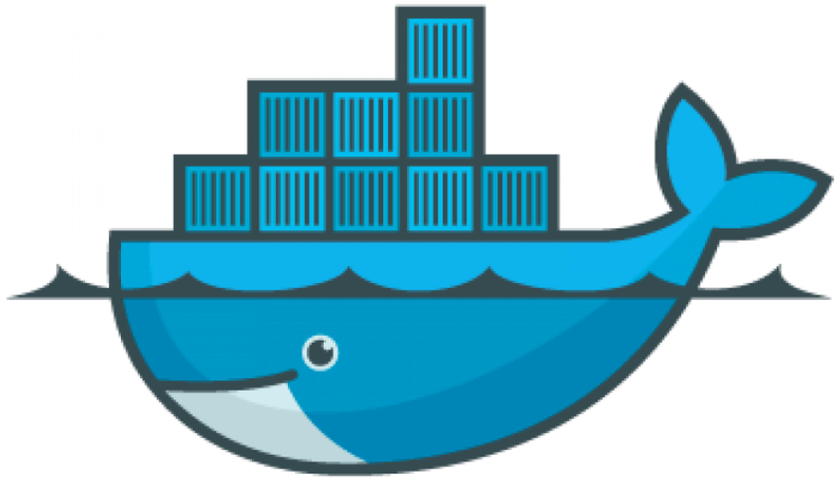
Part 3: Bioinformatic gotchas

Moving between machines

If you do a lot of computation, you may work on several machines.

For example, your laptop, the UBC servers and Compute Canada servers

Trying to make sure that all the programs you are using across these are consistent is a headache



docker

CONDA

https://conda.io/projects/conda/en/latest/user-guide/getting-started.html

Part 3: Bioinformatic gotchas

Also check out this list of resources compiled by UBC postdoc Joey Bernhardt:

https://www.zoology.ubc.ca/~jbernhar/home/?page_id=1836

Good resources on version control, data wrangling and stats

With thanks to:

Jazlyn Mooney

Nichola Hawkins

Brandon Lind

Kieran Samuk

Joanna Rifkin

Alex Krohn

Tutorial:

Work through part 2 of the tutorial associated with this session