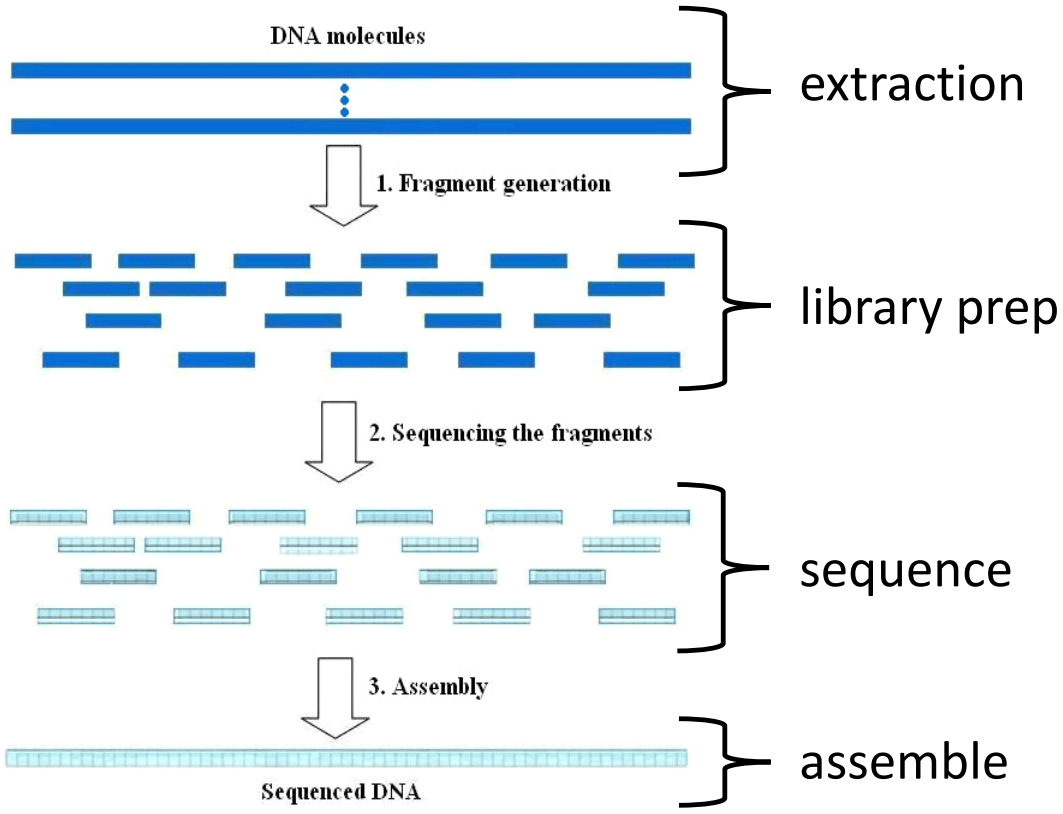
Pre-Processing Your Data

René Clark

Rutgers University

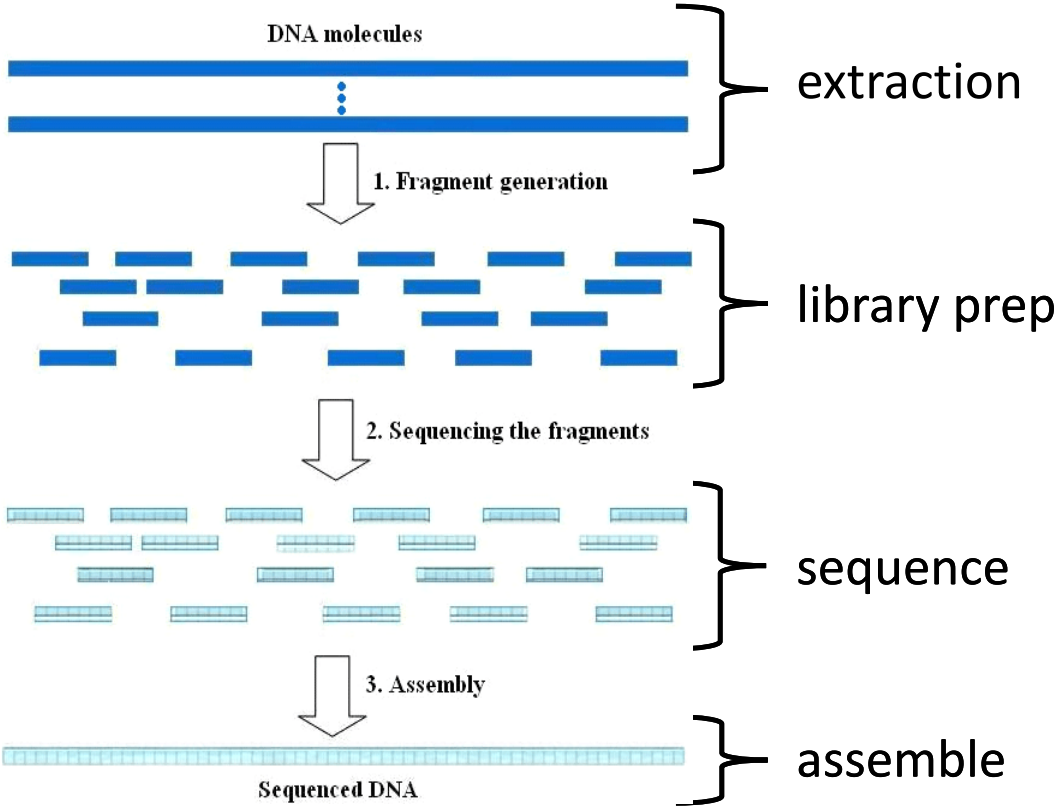
Where did our data come from?

- **Shotgun sequencing:**
  - Randomly shear DNA up into tiny fragments and sequenced them



Where did our data come from?

- **Shotgun sequencing:**
  - Randomly shear DNA up into tiny fragments and sequenced them
- Creates a bunch of “reads” that are collected in a fq.gz file for downstream analysis

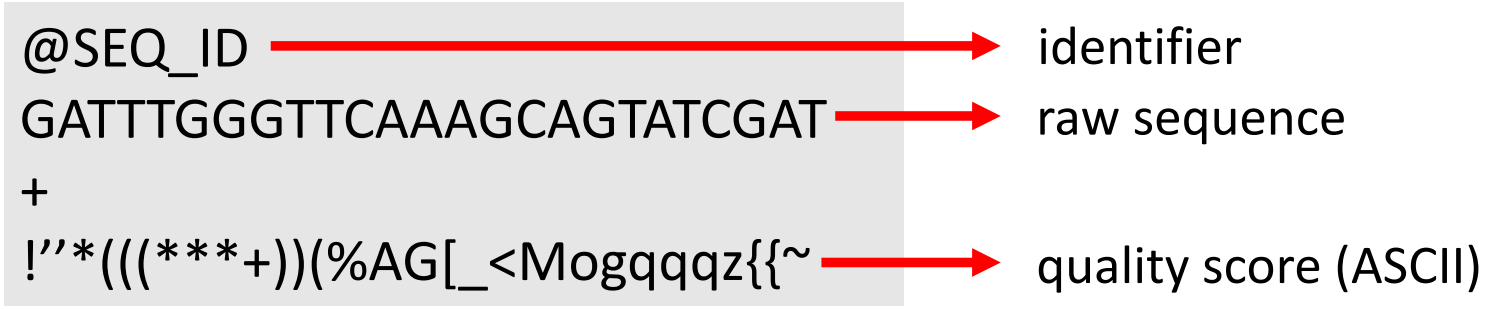


What is a fq.gz file anyway?

- It's the file format you get back from the sequencer!
  - FASTQ
    - gz = zipped (compressed)
- Text-based format for storing both the nucleotide sequence and its corresponding quality scores

What is a fq.gz file anyway?

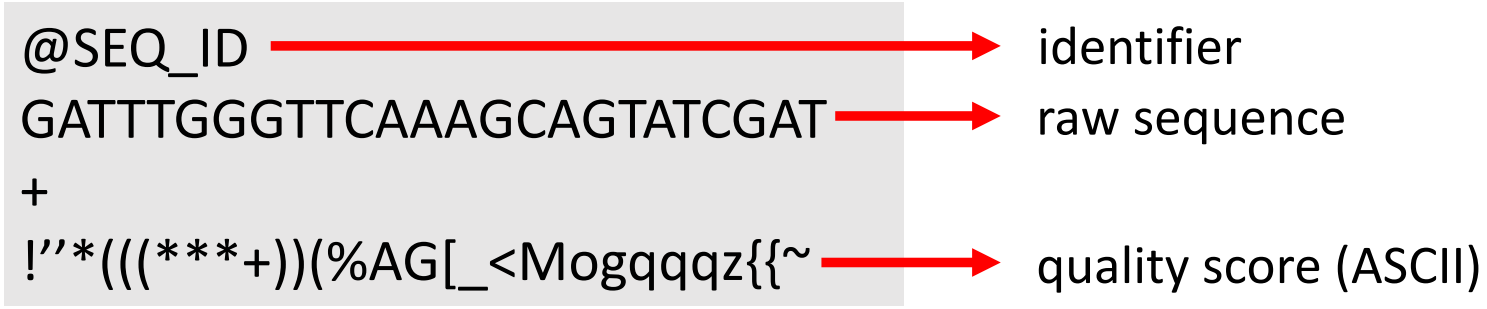
- It's the file format you get back from the sequencer!
  - FASTQ
    - gz = zipped (compressed)
- Text-based format for storing both the nucleotide sequence and its corresponding quality scores



What is a fq.gz file anyway?

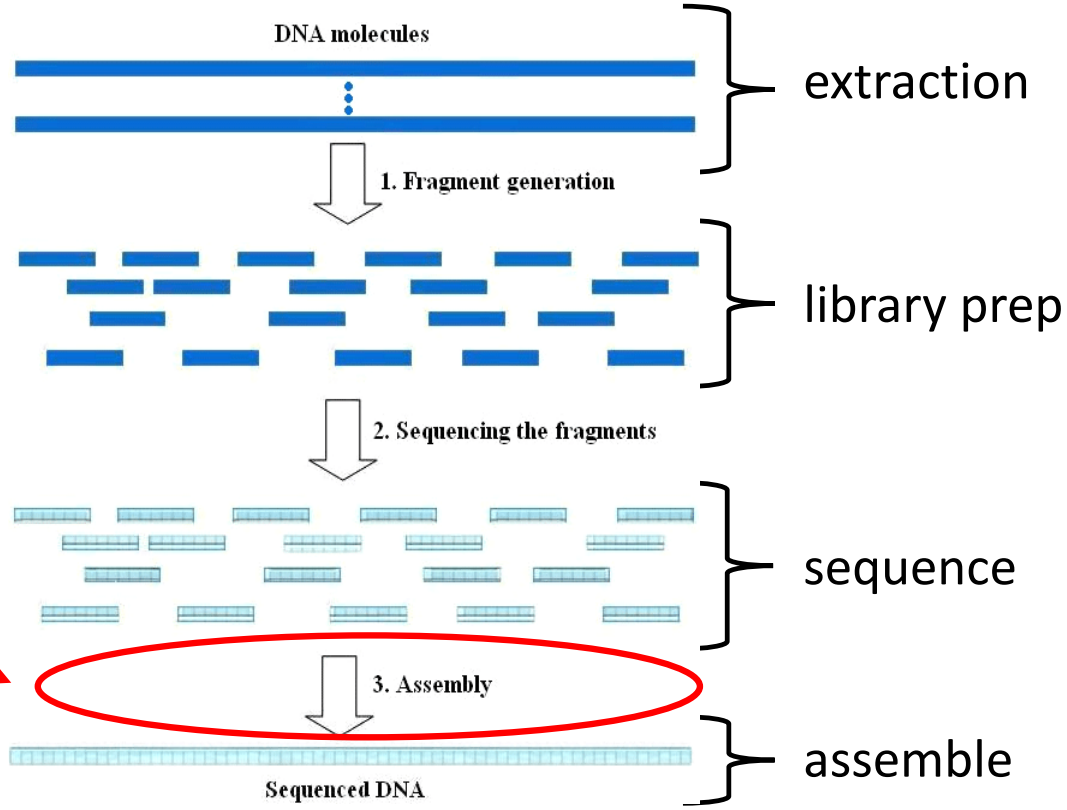
2 files per individual (.R1 & .R2)

- It's the file format you get back from the sequencer!
  - FASTQ
    - gz = zipped (compressed)
- Text-based format for storing both the nucleotide sequence and its corresponding quality scores



Order of operations

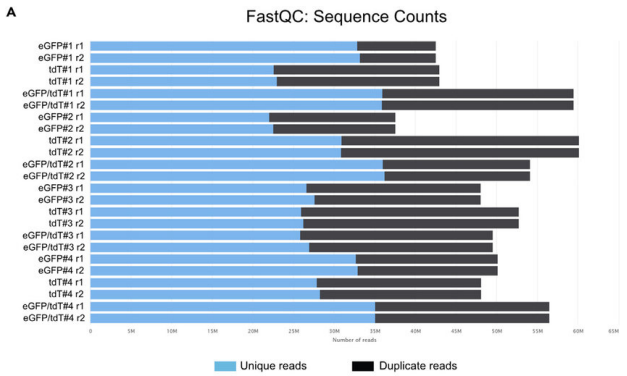
- Before we can assemble, we need to do some quality control:
  1. Check quality of raw data
  2. Trim ends of reads & remove adaptors
  3. De-duplicate reads
  4. Trim beginning of reads
  5. Remove contamination
  6. Re-pair reads



1. Quality assessment (MultiQC/FastQC)

- First, <u>we need to get a “baseline” assessment of our data quality/quantity</u>
  - MultiQC & FastQC help us do that
    - designed to give a brief assessment of sequence quality and identify any “problem areas”

- **FastQC:** generates the report for each fq.gz file
- **MultiQC:** aggregates the information into one summary file



MultiQC

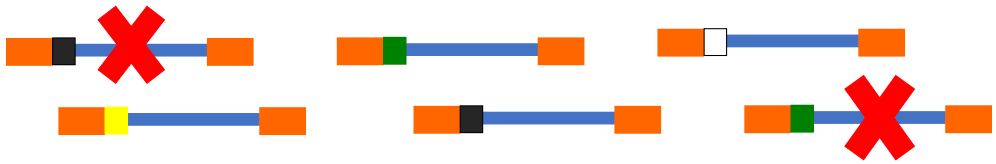
2. Trimming (fastp)

Next, we need to begin **removing low quality bases & "bad" reads**

- 1st trimming:

2. Trimming (fastp)

Next, we need to begin **removing low quality bases & "bad" reads**

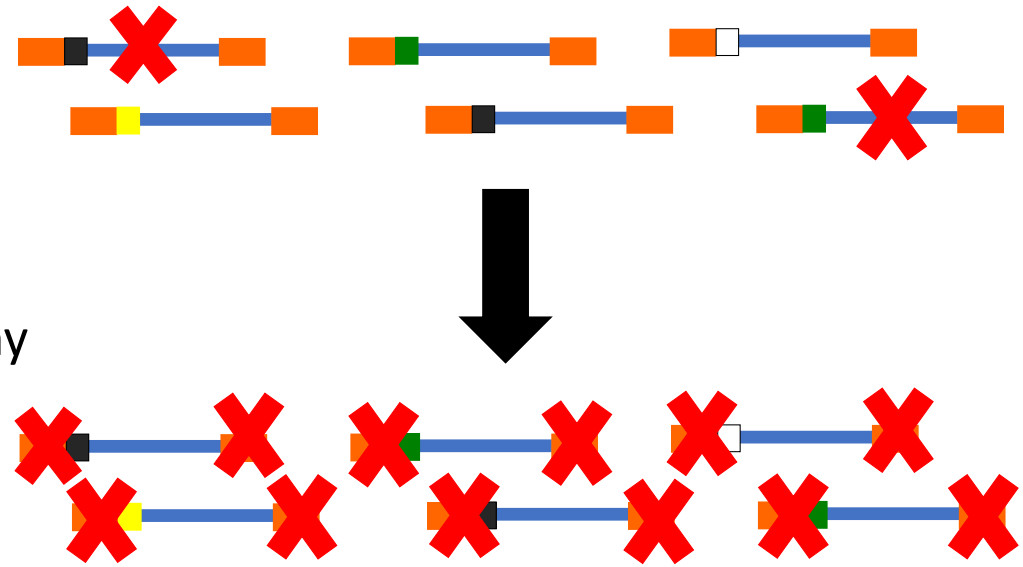


- 1st trimming:
  - Remove reads that are too short
  - Remove reads that have too many low-quality bases

2. Trimming (fastp)

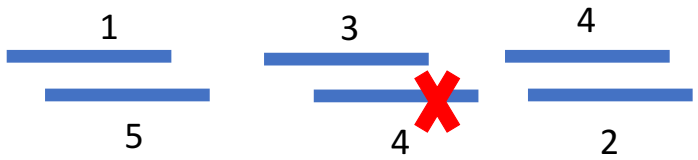
Next, we need to begin **removing low quality bases & "bad" reads**

- 1st trimming:
  - Remove reads that are too short
  - Remove reads that have too many low-quality bases
  - Remove 3' "ends" of reads
  - Remove adaptor sequences



3. De-duplicating (clumpify)

- Now, we want to **reduce data size** to make the assembly more efficient & to minimize effects of sequencing errors
- Can get "duplicate" reads due to PCR amplification during the library prep stage
  - We want to remove these reads, to reduce risk of perpetuating sequencing errors



3. De-duplicating (clumpify)

- Now, we want to **reduce data size** to make the assembly more efficient & to minimize effects of sequencing errors
- Can get "duplicate" reads due to PCR amplification during the library prep stage
  - We want to remove these reads, to reduce risk of perpetuating sequencing errors

Clumpify will sort reads in fq.gz into "clumps"

Order them by similarity and remove duplicates

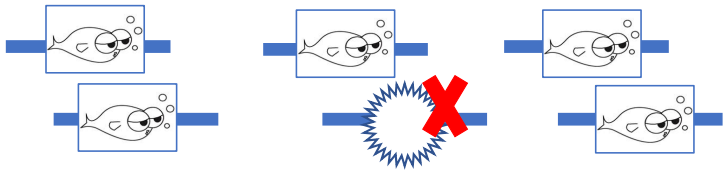
4. Trimming round 2 (fastp)

- Next, we trim our reads again

- This time, we are trimming the beginning (5”) end of the reads if they have low quality bases

5. Removing any contamination (FastQ Screen)

- **<u>What if our data has non-fish DNA????</u>**
- Want to remove any contamination (non-target DNA) that may have occurred during sample storage, extraction, or library prep
  - *If this is left in, it could screw up our assembly downstream!!*
- FastQ Screen searches our DNA against a database of other genomes to identify (and remove) extraneous DNA

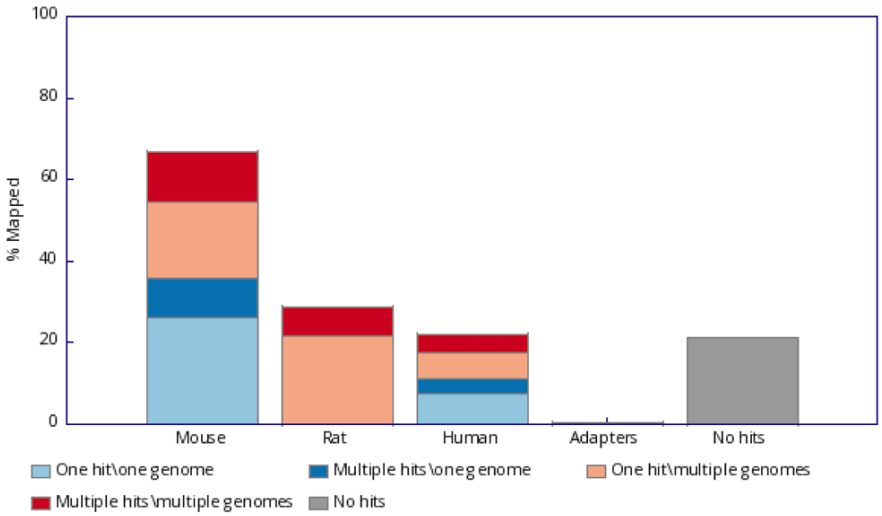


5. Removing any contamination (FastQ Screen)

- **<u>What if our data has non-fish DNA????</u>**

- Want to remove any contamination (non-target DNA) that may have occurred during sample storage, extraction, or library prep
  - *If this is left in, it could screw up our assembly downstream!!*

- FastQ Screen searches our DNA against a database of other genomes to identify (and remove) extraneous DNA



6. Re-pair reads

- Finally, we need to re-pair any read pairs that got separated during the previous steps

- Ensures the R1 & R2 fq.gz files are in the same order
  - Makes assembly, etc. go a lot more smoothly!

```
breid@e3-w6420b-01:/home/r3clark/PIRE/2022_PIRE_omics_workshop/test_student/shotgun_raw_fq$ bash /home/elgarcia/shotgun_PIRE/pire_fq_gz_processing/renameFQGZ.bash Sfa_ProbeDevelopmentLibraries_SequenceNameDecode.tsv rename

decode file read into memory
rename specified, files will be renamed
Are you sure? y
bash renameFQGZ.bash Sfa_ProbeDevelopmentLibraries_SequenceNameDecode.tsv rename
writing original file names to file, origFileNames.txt...
writing newFileNames.txt...
editing newFileNames.txt...
preview of orig and new R1 file names...
SfC0281G_CKDL210013395-1a-AK3911-AK845_HF33GDSX2_L4_1.fq.gz Sfa-CBas_028-Ex1-1G_L4_1.fq.gz
SfC0281H_CKDL210013395-1a-5UDI245-GD07_HF33GDSX2_L4_1.fq.gz Sfa-CBas_028-Ex1-1H_L4_1.fq.gz
SfC0282A_CKDL210013395-1a-AK8593-7UDI304_HF33GDSX2_L4_1.fq.gz Sfa-CBas_028-Ex1-2A_L4_1.fq.gz
preview of orig and new R2 file names...
SfC0281G_CKDL210013395-1a-AK3911-AK845_HF33GDSX2_L4_2.fq.gz Sfa-CBas_028-Ex1-1G_L4_2.fq.gz
SfC0281H_CKDL210013395-1a-5UDI245-GD07_HF33GDSX2_L4_2.fq.gz Sfa-CBas_028-Ex1-1H_L4_2.fq.gz
SfC0282A_CKDL210013395-1a-AK8593-7UDI304_HF33GDSX2_L4_2.fq.gz Sfa-CBas_028-Ex1-2A_L4_2.fq.gz

Last chance to back out. If the original and new file names look ok, then proceed.
Are you sure you want to rename the files? y
renaming R1 files...
renaming R2 files...
breid@e3-w6420b-01:/home/r3clark/PIRE/2022_PIRE_omics_workshop/test_student/shotgun_raw_fq$
```

3. De-duplicating (clumpify)

- Now, we want to **reduce data size** to make the assembly more efficient & to minimize effects of sequencing errors
- Can get "duplicate" reads due to PCR amplification during the library prep stage
  - We want to remove these reads, to reduce risk of perpetuating sequencing errors