SPAAM

Standards, Precautions & Advances in Ancient Metagenomics

Practical 2D: Introduction to nf-core/eager

Megan Michel & James A. Fellows Yates

Overview

Before we start!

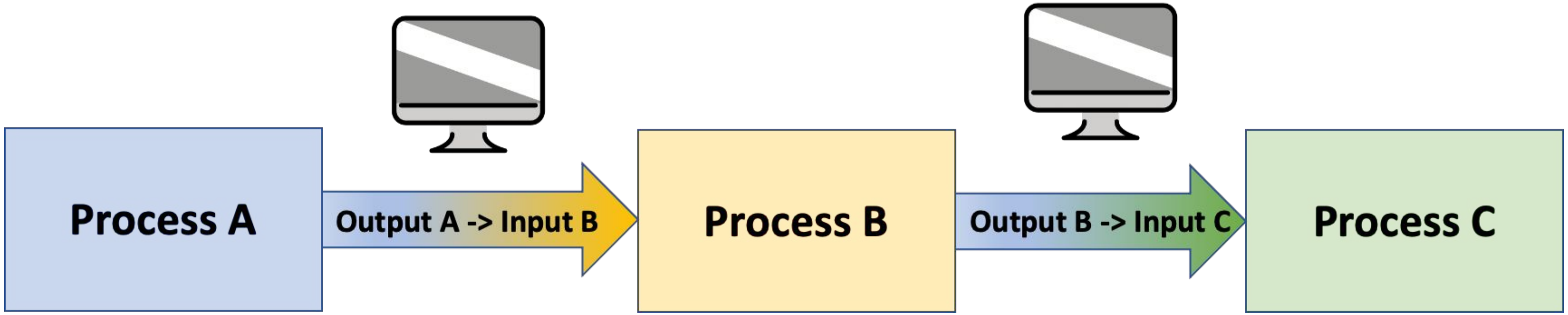
```
$ cd /vol/volume
$ ls
```

If you see **only** lost+found please run the following
(if you see other directories, you're 👍)

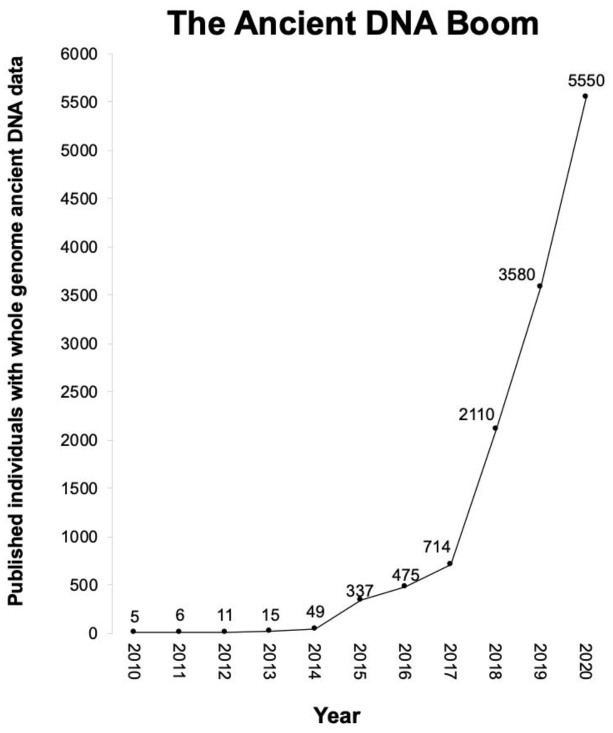
```
$ wget https://share.eva.mpg.de/index.php/s/Go8DH44JYFSrLXj/download -O
2d-introduction-to-nf-core-eager.tar
$ tar xvf *eager.tar
$ rm *eager.tar
```

What is a pipeline?

- Series of linked computational steps in which the output of one process becomes the input to the next

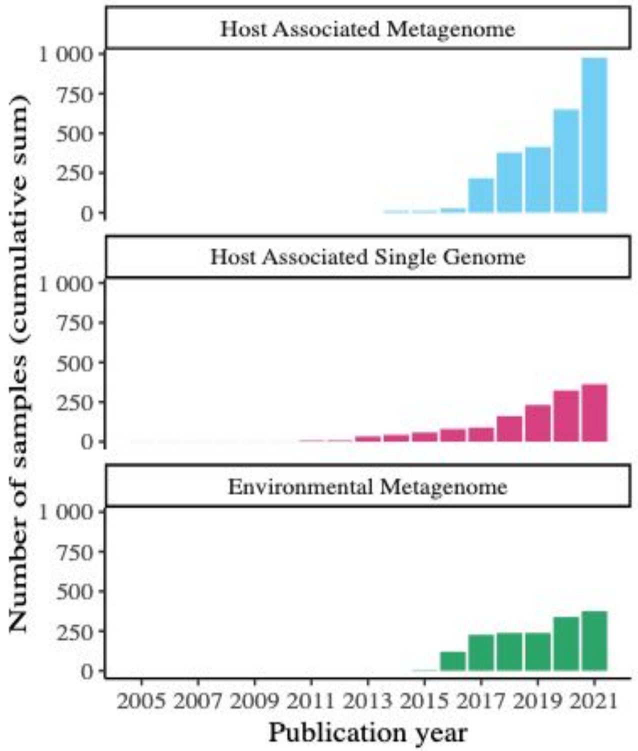


More Data, More Analyses



Ancient Human Genomes*

https://reich.hms.harvard.edu/research

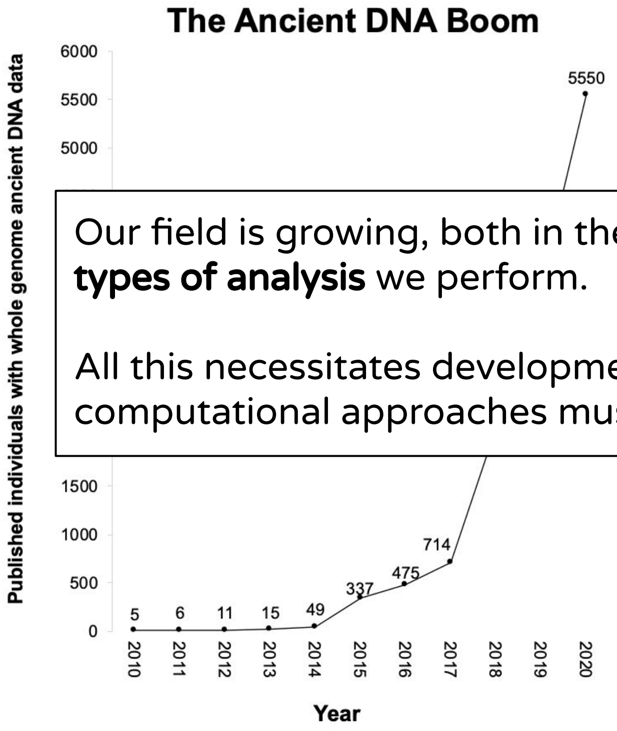


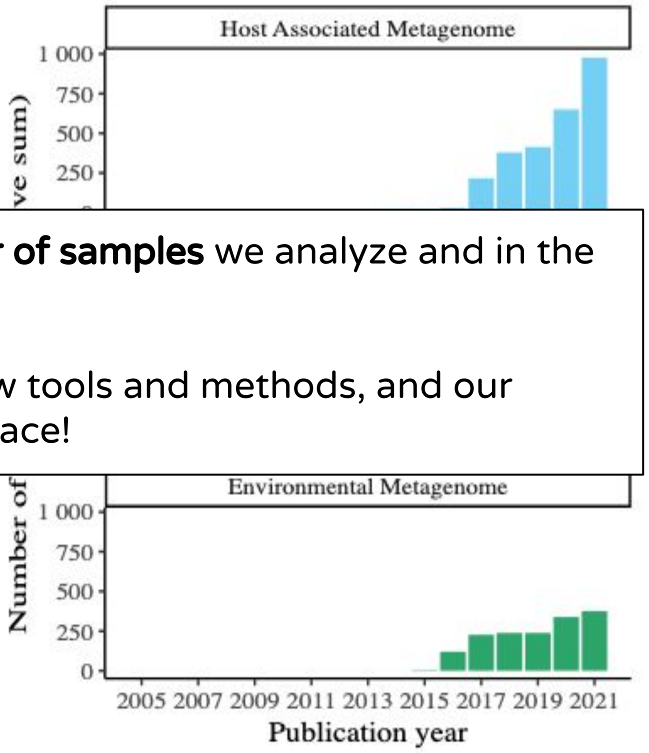
Other aDNA Sources

https://github.com/SPAAM-community/AncientMetagenomeDir

*Genome wide data

More Data, More Analyses





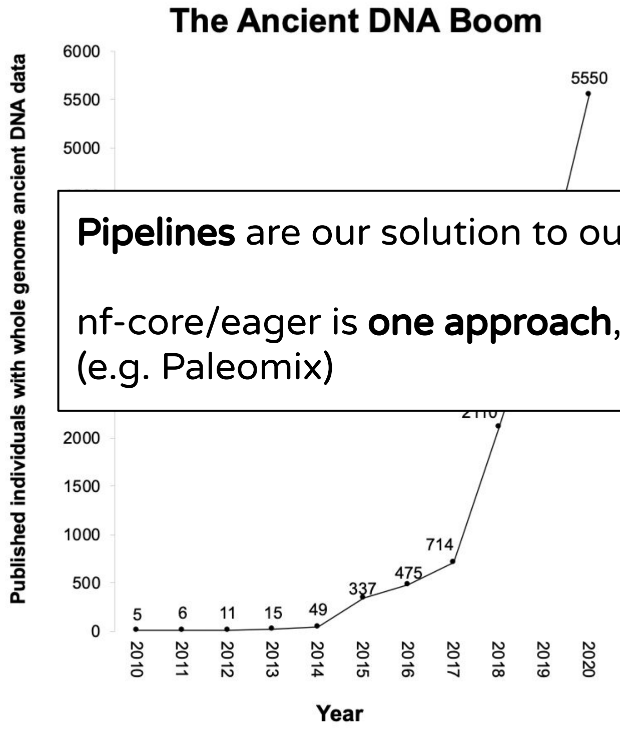
Our field is growing, both in the **number of samples** we analyze and in the **types of analysis** we perform.

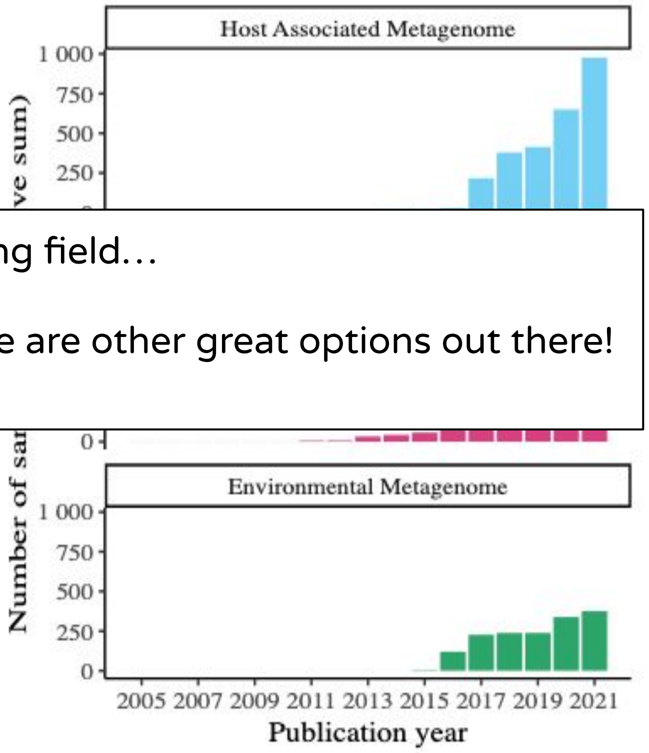
All this necessitates development of new tools and methods, and our computational approaches must keep pace!

Ancient Human Genomes
https://reich.hms.harvard.edu/research

Other aDNA Sources
https://github.com/SPAAM-community/AncientMetagenomeDir

More Data, More Analyses





Pipelines are our solution to our changing field…

nf-core/eager is **one approach**, but there are other great options out there! (e.g. Paleomix)

Ancient Human Genomes
https://reich.hms.harvard.edu/research

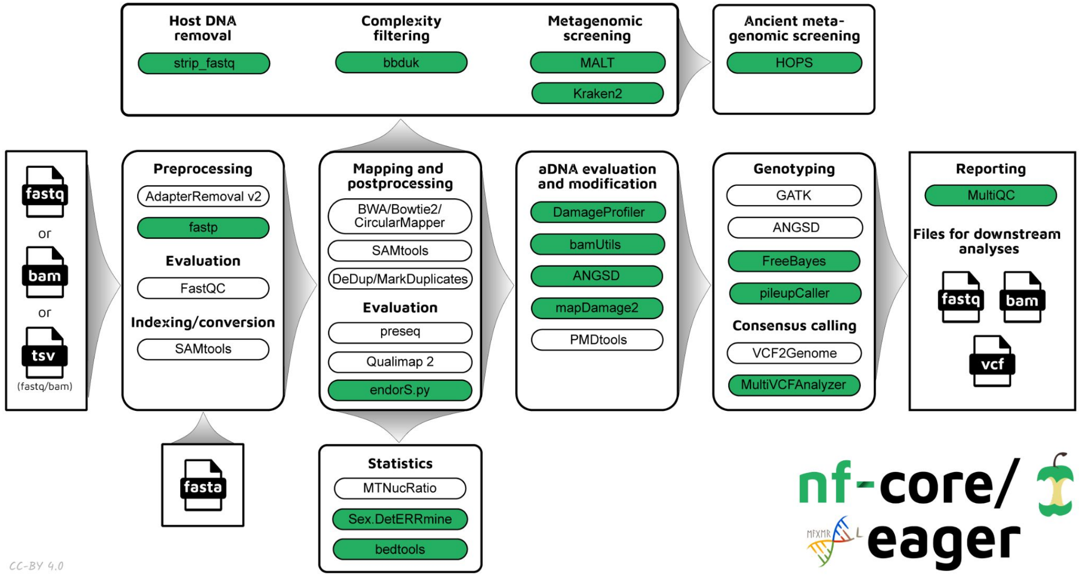
Other aDNA Sources
https://github.com/SPAAM-community/AncientMetagenomeDir

Introduction to nf-core/eager

What is nf-core/eager?

- **Computational pipeline** designed for **ancient DNA data**
  - **EAGER**- Efficient Ancient Genome Reconstruction (Peltzer *et al.* 2016)
- nf-core/eager- Reimplementation of the pipeline using **Nextflow**

1. **Portability**
2. **Reproducibility**
3. **Updated functionality**



nf-core/eager

Portability/Accessibility

- **Why do we care about portability?**
  - Facilitates **reproducible analyses** by ensuring same tool versions, dependencies, environment, etc.
  - Provides **easy access to pipeline tools** (fewer installation/dependency issues)
  - Facilitates **use across platforms** (HPC systems, PCs, cloud computing)

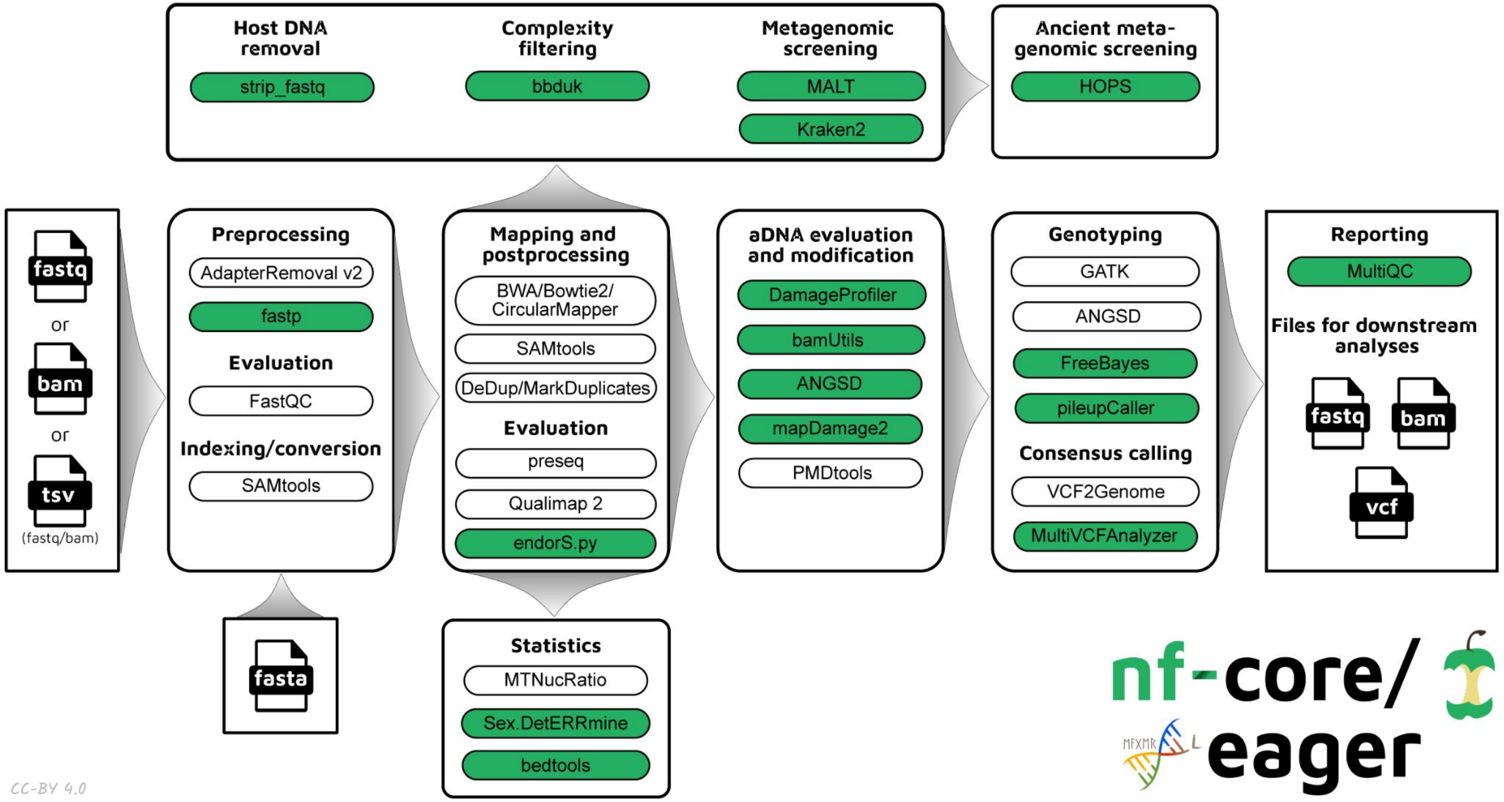
Portability/Accessibility

- **Containerization-** distributing programs in self-contained bundles that contain all the code, packages, libraries, etc. needed to run it
- nf-core/eager compatible with **Docker**, **Conda**, **Singularity**

Reproducibility

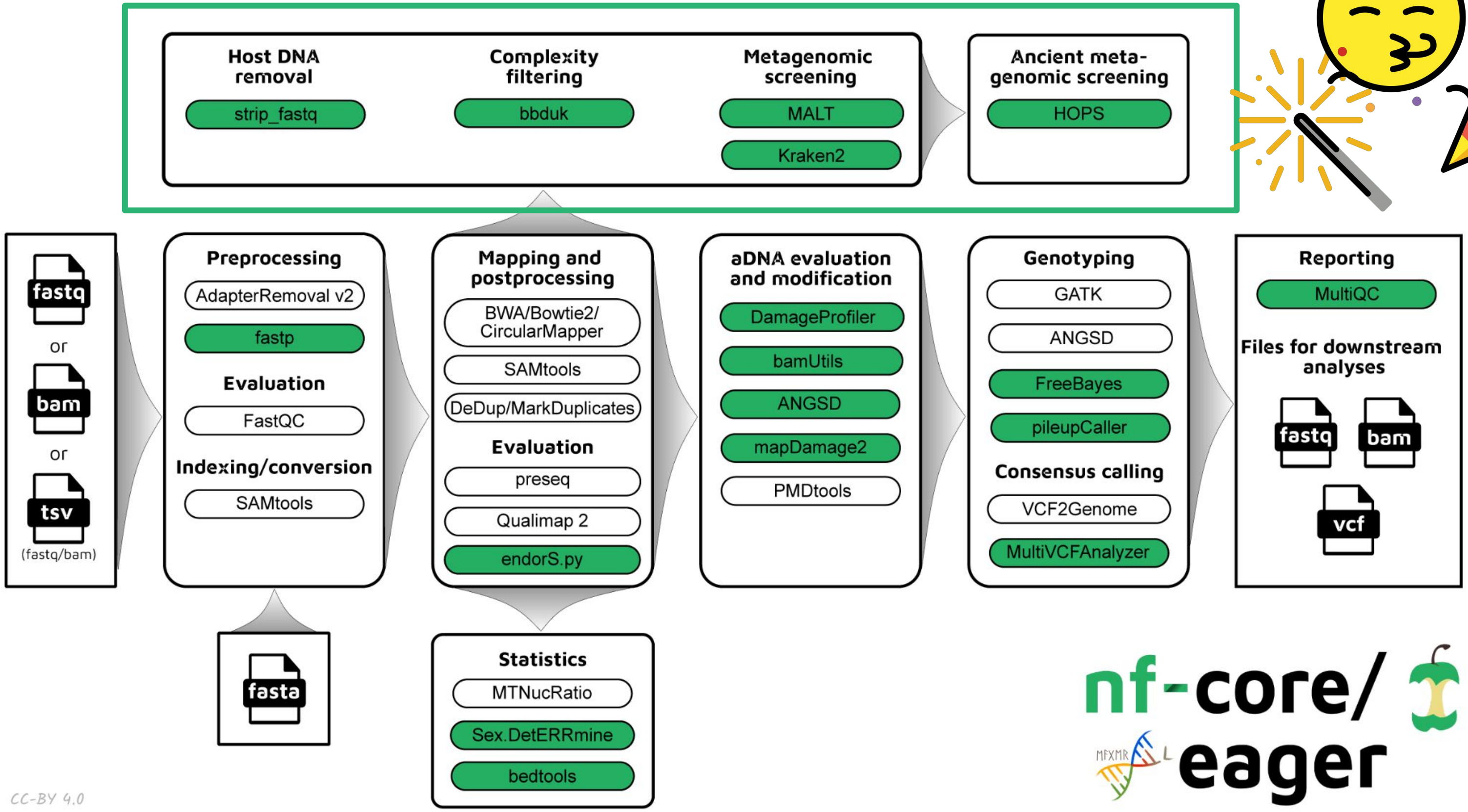
- **Customizable configuration profiles**:
    - *HPC cluster-level profiles*- usable with all nextflow pipelines!
    - *Pipeline-level profiles*- Specifies analytical options for easily-reproducible analyses
- Profiles can be shared alongside publications (e.g. via Github)
    - Easier to write your methods section

New pipeline, new tools!



nf-core/eager

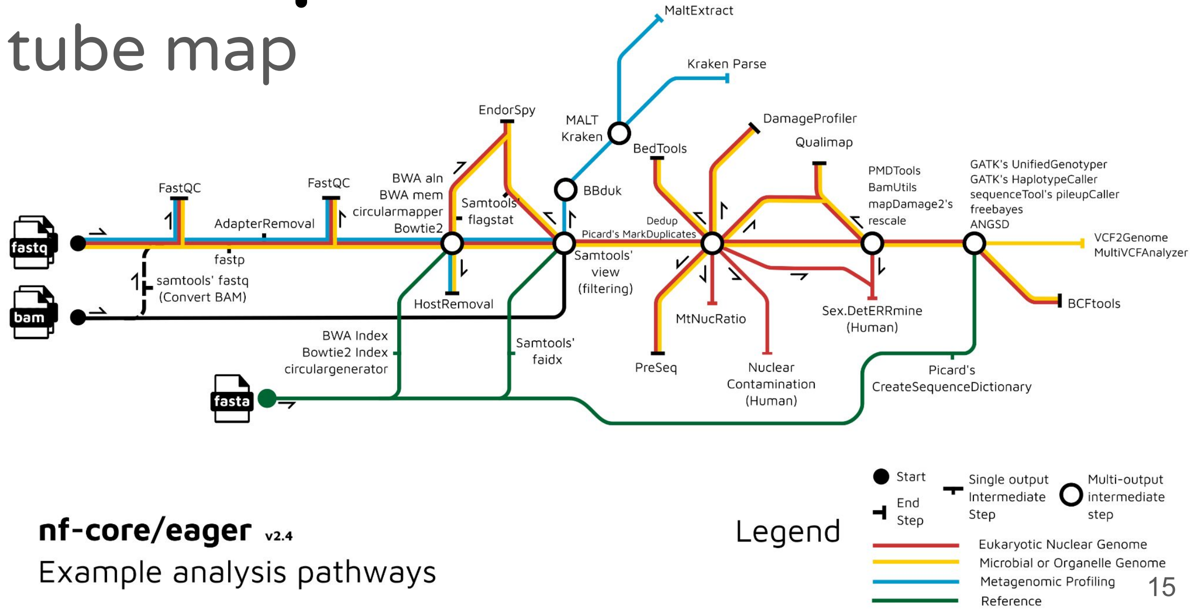
New pipeline, new tools!

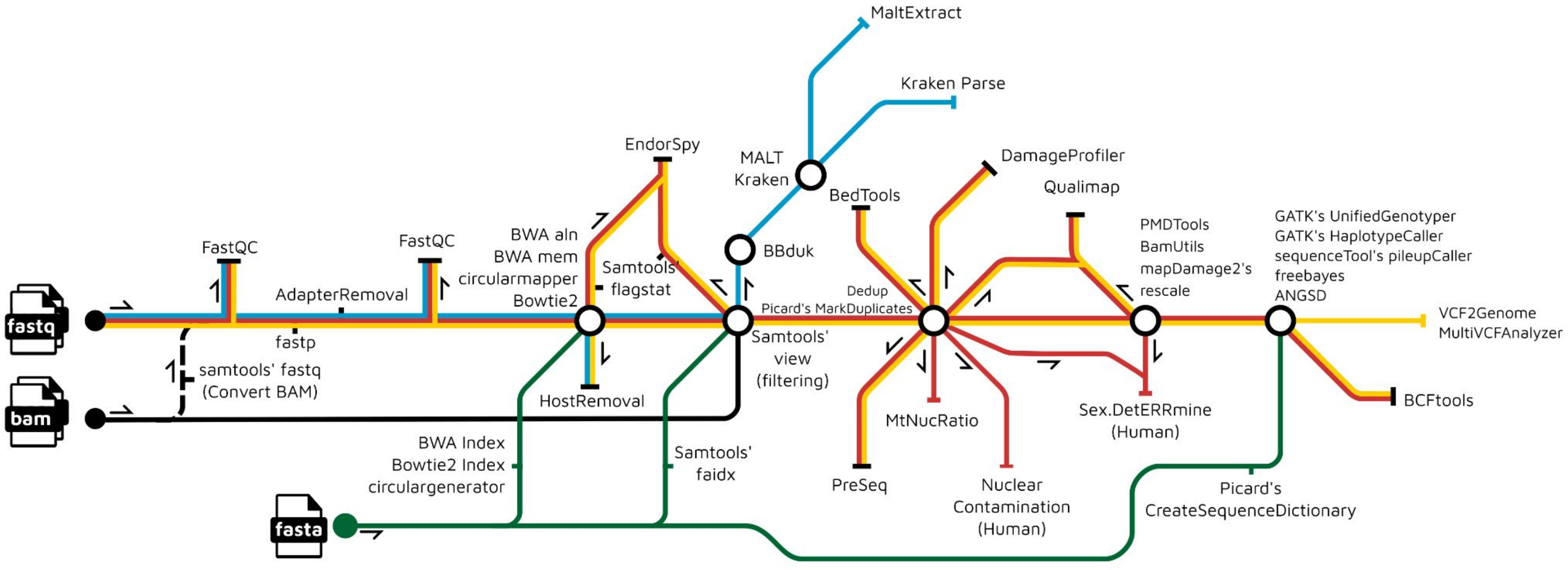


nf-core/eager

Steps in the Pipeline

Or how to read the tube map





nf-core/eager v2.4

Example analysis pathways

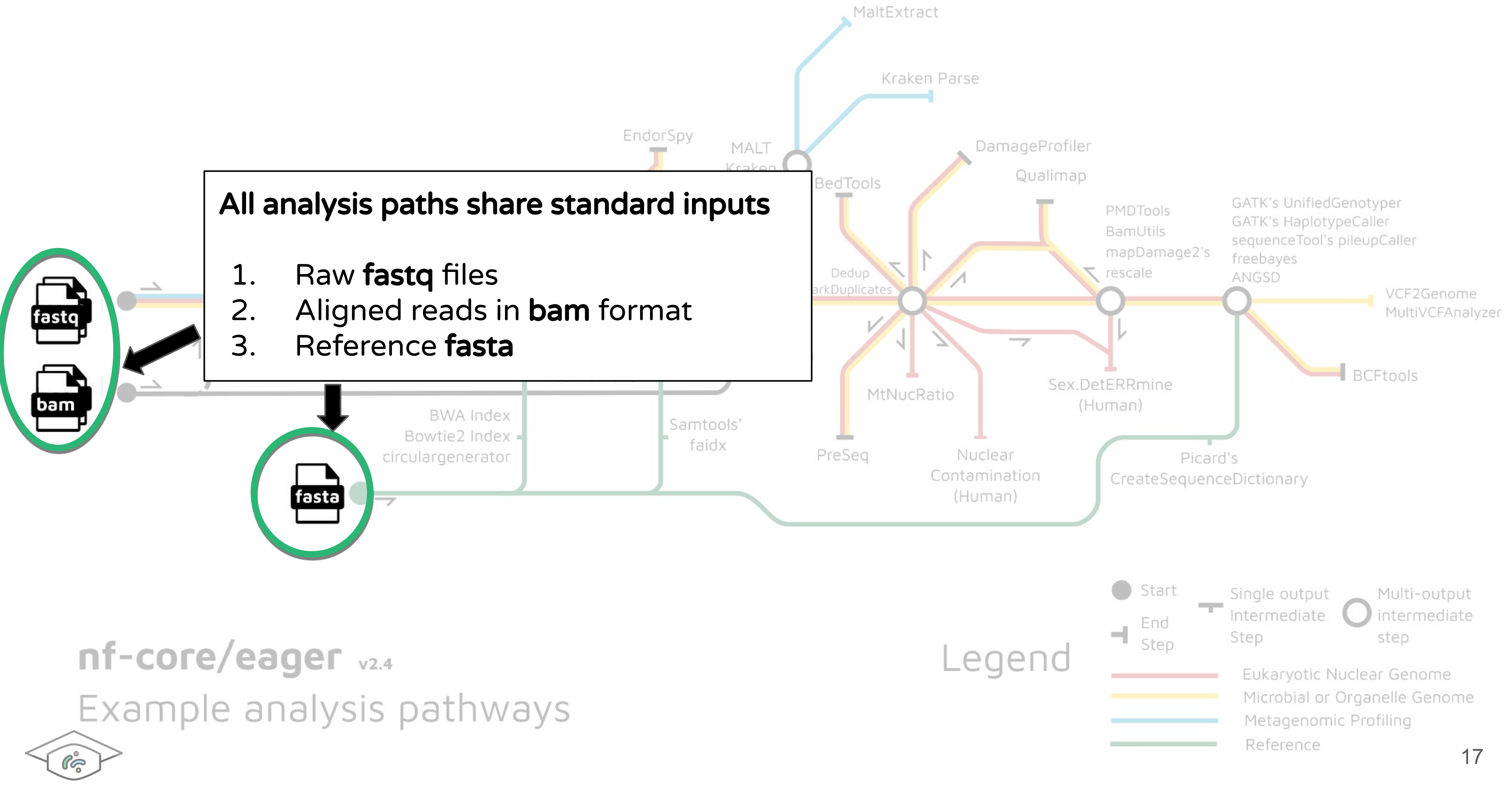
Legend

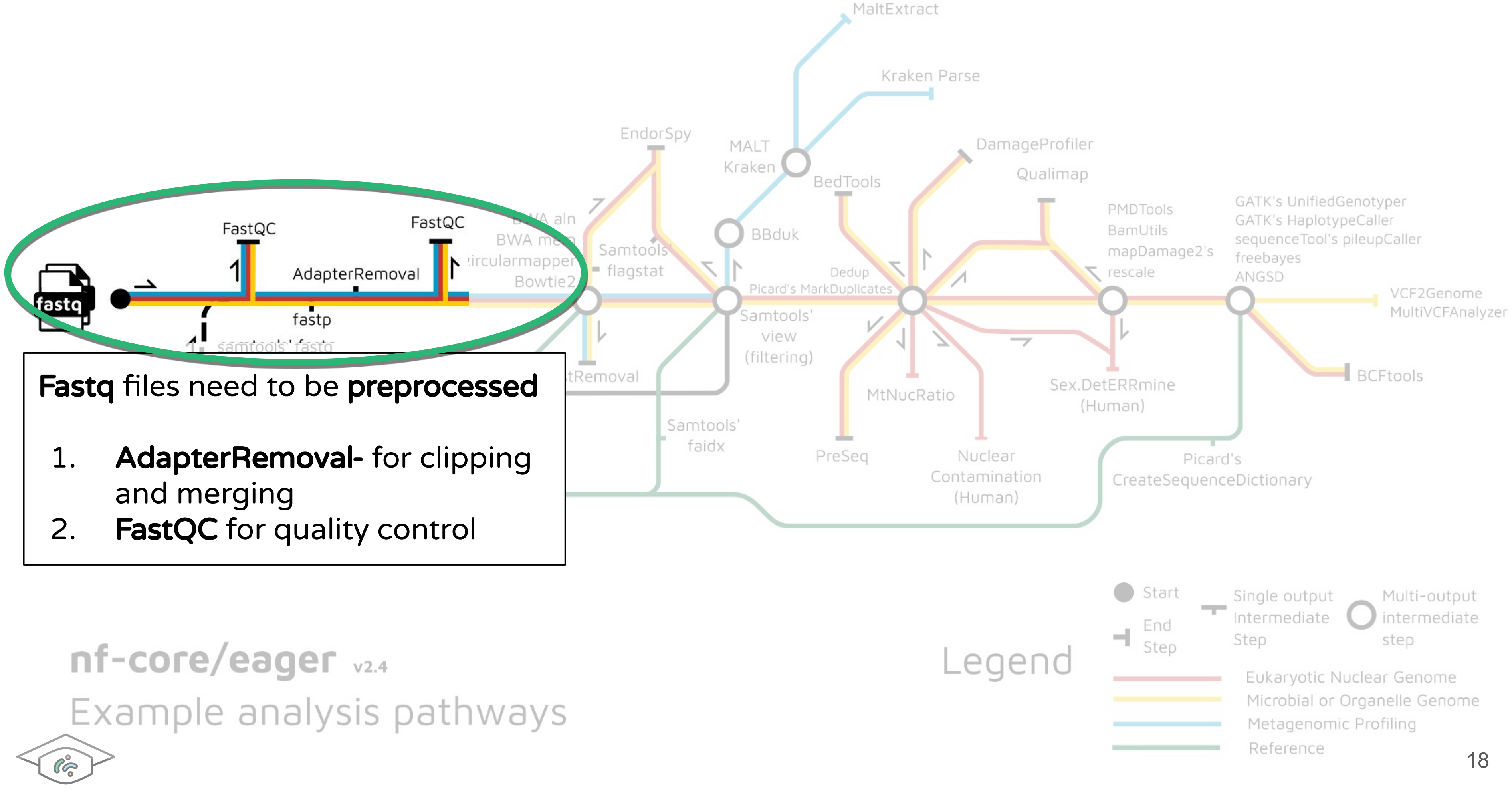
●	Start	┳	Single output Intermediate Step
┫	End Step	○	Multi-output intermediate step

🟥	Eukaryotic Nuclear Genome
🟨	Microbial or Organelle Genome
🟦	Metagenomic Profiling
🟩	Reference

All analysis paths share standard inputs

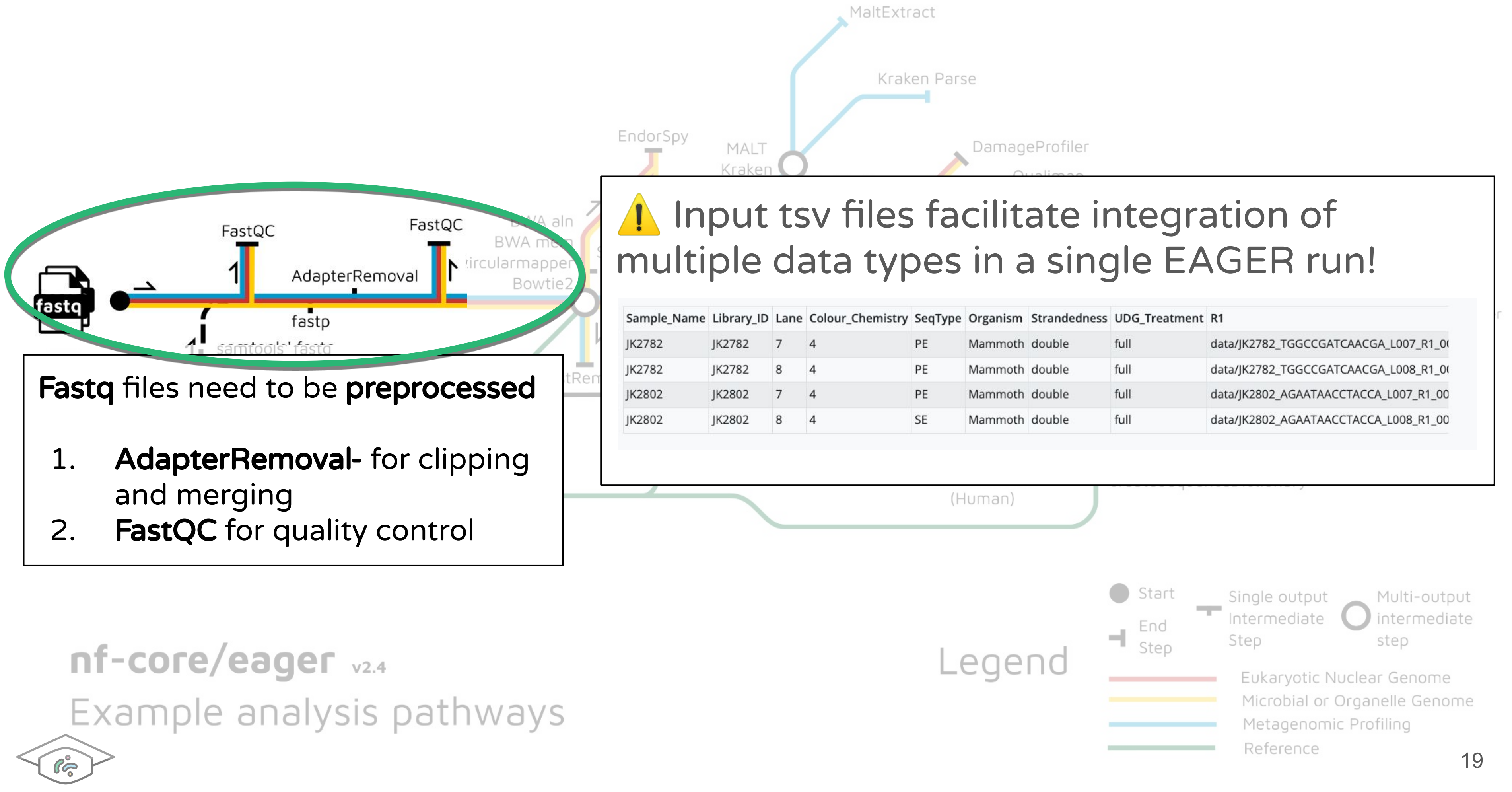
1. Raw **fastq** files
2. Aligned reads in **bam** format
3. Reference **fasta**





Fastq files need to be **preprocessed**

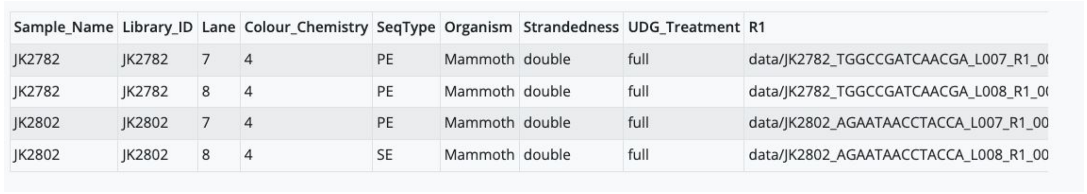
1. **AdapterRemoval-** for clipping and merging
2. **FastQC** for quality control



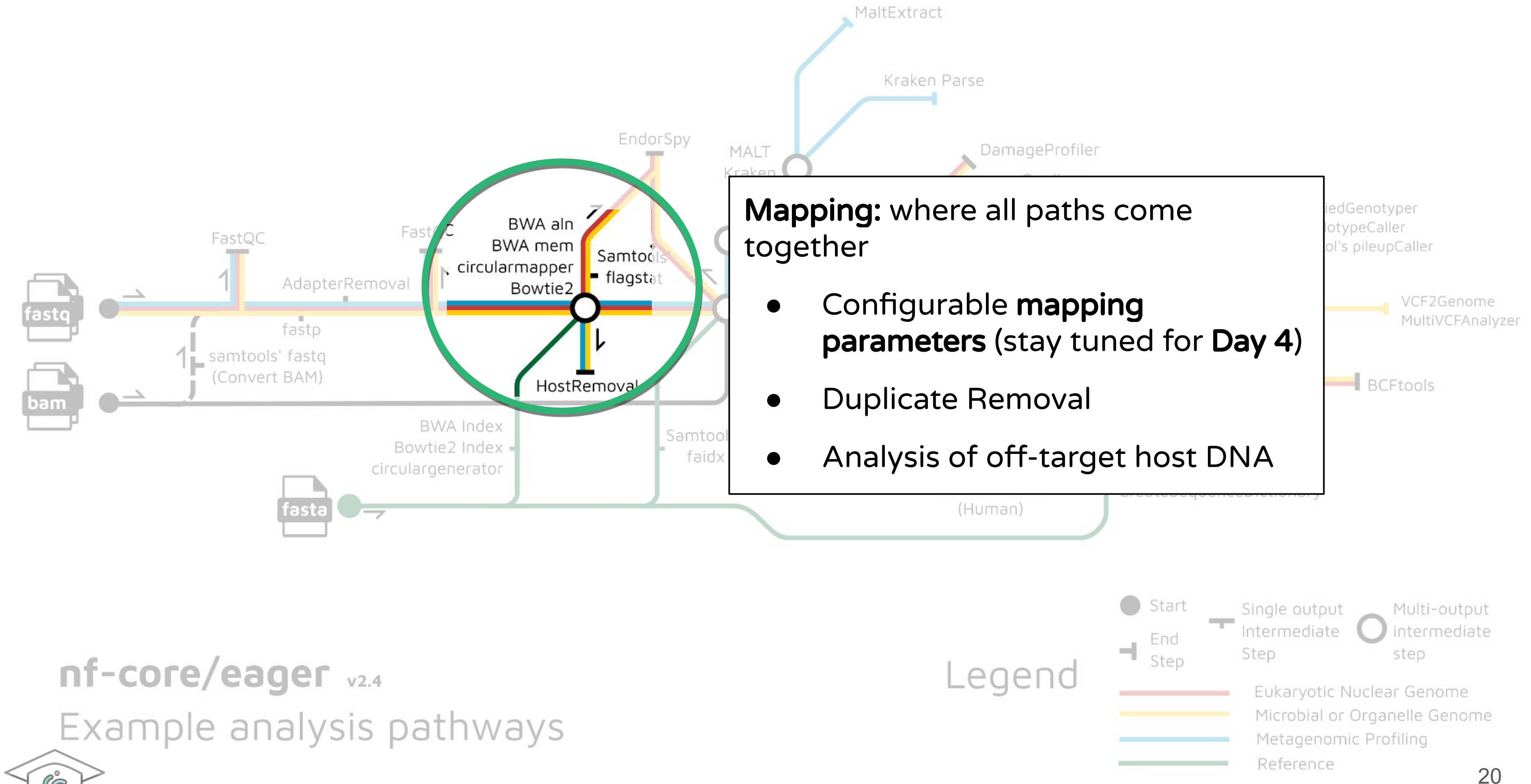
Fastq files need to be **preprocessed**

1. **AdapterRemoval-** for clipping and merging
2. **FastQC** for quality control

⚠️ Input tsv files facilitate integration of multiple data types in a single EAGER run!

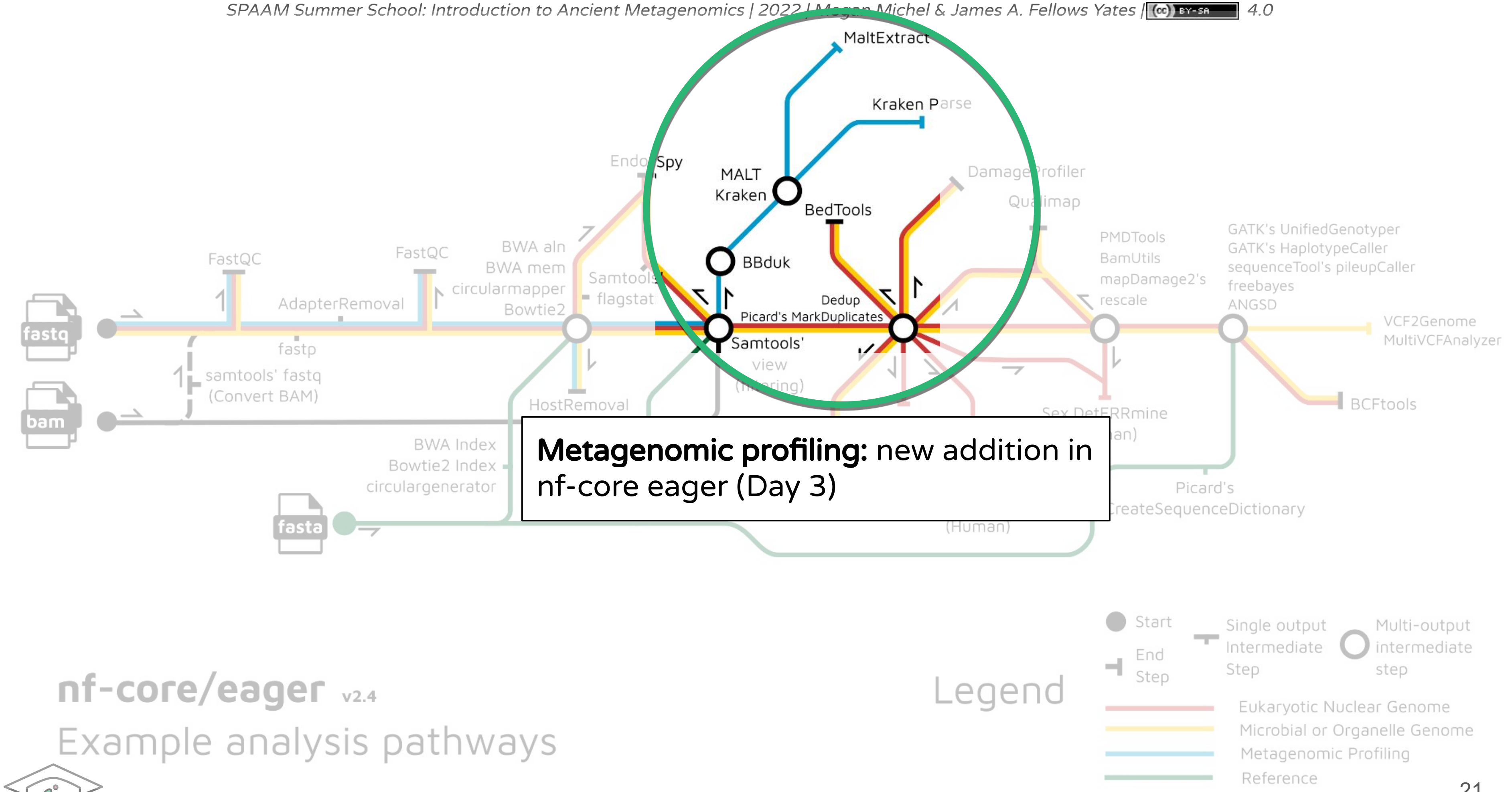


Sample_Name	Library_ID	Lane	Colour_Chemistry	SeqType	Organism	Strandedness	UDG_Treatment	R1
JK2782	JK2782	7	4	PE	Mammoth	double	full	data/JK2782_TGGCCGATCAACGA_L007_R1_0
JK2782	JK2782	8	4	PE	Mammoth	double	full	data/JK2782_TGGCCGATCAACGA_L008_R1_0
JK2802	JK2802	7	4	PE	Mammoth	double	full	data/JK2802_AGAATAACCTACCA_L007_R1_00
JK2802	JK2802	8	4	SE	Mammoth	double	full	data/JK2802_AGAATAACCTACCA_L008_R1_00

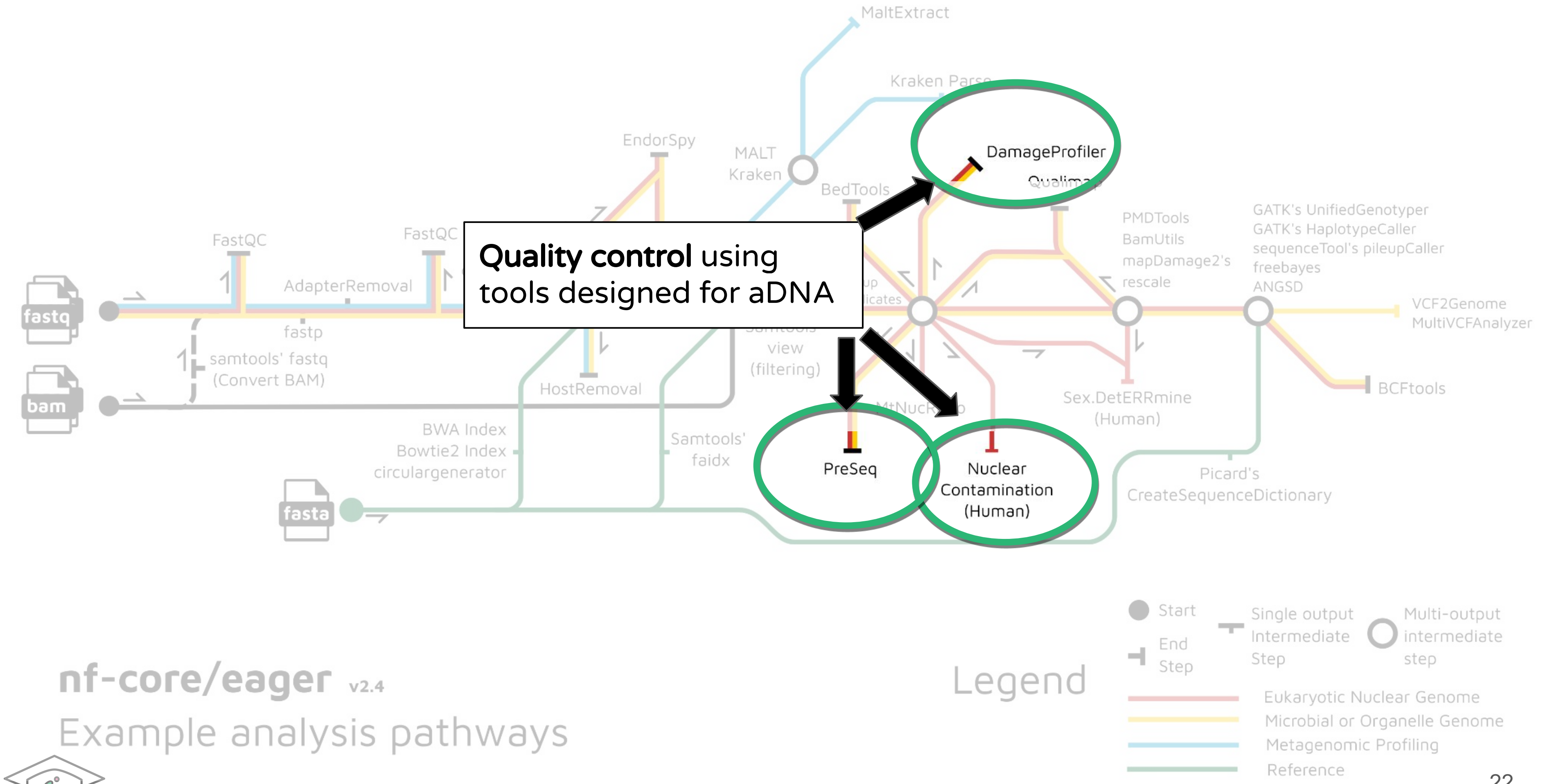


Mapping: where all paths come together

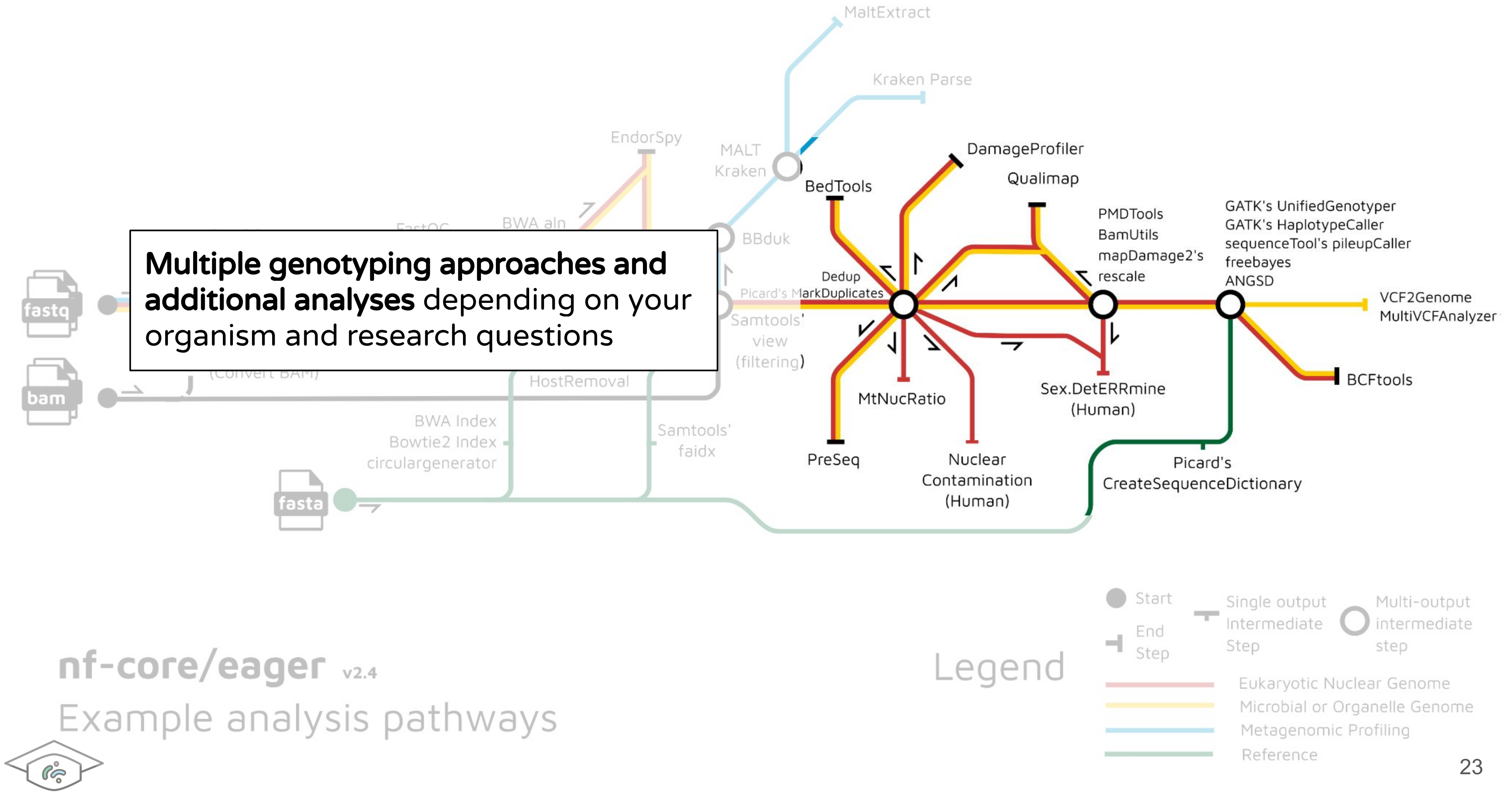
- Configurable **mapping parameters** (stay tuned for **Day 4**)
- Duplicate Removal
- Analysis of off-target host DNA



Metagenomic profiling: new addition in nf-core eager (Day 3)



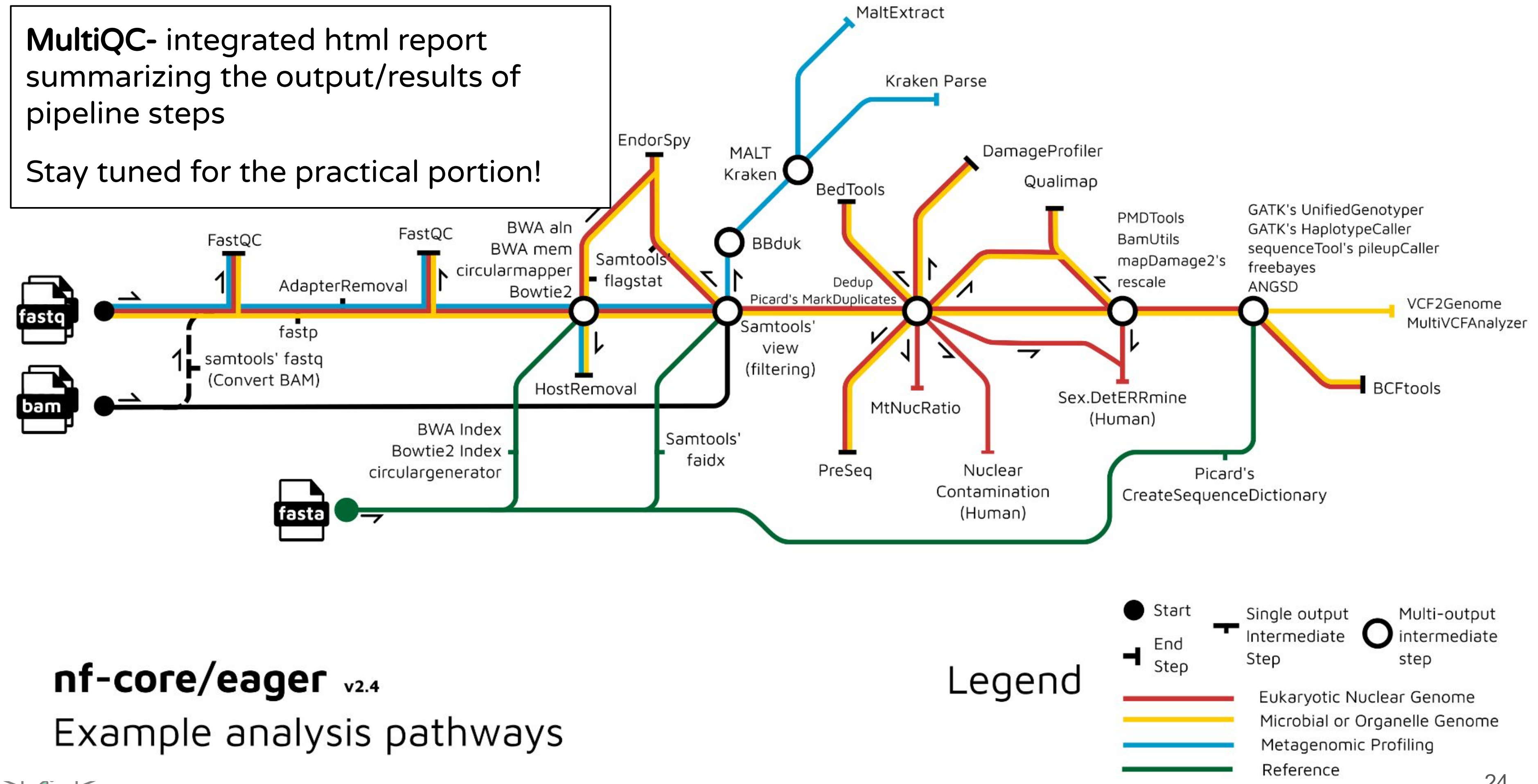
Quality control using tools designed for aDNA



Multiple genotyping approaches and additional analyses depending on your organism and research questions

MultiQC- integrated html report summarizing the output/results of pipeline steps

Stay tuned for the practical portion!



How to build an EAGER command

A practical introduction

Our dataset: *Y. pestis* capture

Sample Name	Library Name	Strand Type	Library Treatment	Instrument Model	Library Layout	Library Strategy	Archive Data Accession
GLZ002	GLZ002.A0101	double	half-udg	Illumina HiSeq 4000	PAIRED	Targeted-Capture	ERR4093961
GLZ002	GLZ002.A0102	double	full-udg	Illumina HiSeq 4000	PAIRED	Targeted-Capture	ERR4093962
KZL002	KZL002.A0101	double	half-udg	Illumina HiSeq 4000	PAIRED	Targeted-Capture	ERR8958768
KZL002	KZL002.A0102	double	half-udg	Illumina HiSeq 4000	PAIRED	Targeted-Capture	ERR8958769

- **Glazkovskoe predmestie (GLZ)-** Neolithic Siberia, 3081-2913 calBCE
- **Kyzyl (KZL)-** Iron Age Kazakhstan, 2736-2457 calBCE

H. Yu *et al*. Paleolithic to Bronze Age Siberians Reveal Connections with First Americans and across Eurasia. *Cell*. **181**, 1232–1245.e20 (2020); A. A. Valtueña *et al.* Stone Age *Yersinia pestis* genomes shed light on the early evolution, diversity, and ecology of plague. *Proceedings of the National Academy of Sciences*. **119**, e2116722119 (2022).

Our dataset: *Y. pestis* capture

Sample Name	Library Name	Strand Type	Library Treatment	Instrument Model	Library Layout	Library Strategy	Archive Data Accession
GLZ002	GLZ002.A0101	double	half-udg	Illumina HiSeq 4000	PAIRED	Targeted-Capture	ERR4093961
GLZ002	GLZ002.A0102	double	full-udg	Illumina HiSeq 4000	PAIRED	Targeted-Capture	ERR4093962
KZL002	KZL002.A0101	double	half-udg	Illumina HiSeq 4000	PAIRED	Targeted-Capture	ERR8958768
KZL002	KZL002.A0102	double	half-udg	Illumina HiSeq 4000	PAIRED	Targeted-Capture	ERR8958769

Our target:

1. **Trim adapters** and **merge** raw sequencing data
2. **Align** reads to the *Y. pestis* reference and **compute endogenous percent**
3. Filter bam files to **remove host DNA**
4. **Deduplicate** reads for accurate coverage estimation and genotyping
5. **Merge data** by sample and perform **genotyping** on combined dataset

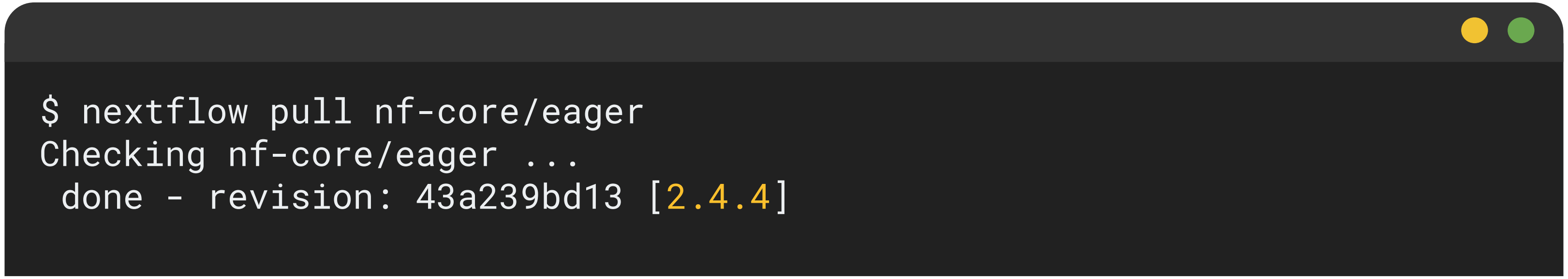
On your marks…

- **Conda environment** contains pre-installed tools for this practical session
  - ⚠️ For future installation (e.g. on your local machine), see: https://nf-co.re/eager

```
$ conda activate git-eager
$ cd /vol/volume/2d-introduction-to-nf-core-eager/eager
```

Get Set(up)...

Download the **latest version** of the nf-core/eager repo (or **update** an already installed version)



```
$ nextflow pull nf-core/eager
Checking nf-core/eager ...
 done - revision: 43a239bd13 [2.4.4]
```

Go run nf-core/eager!

Don't press enter until we make it through the whole command!

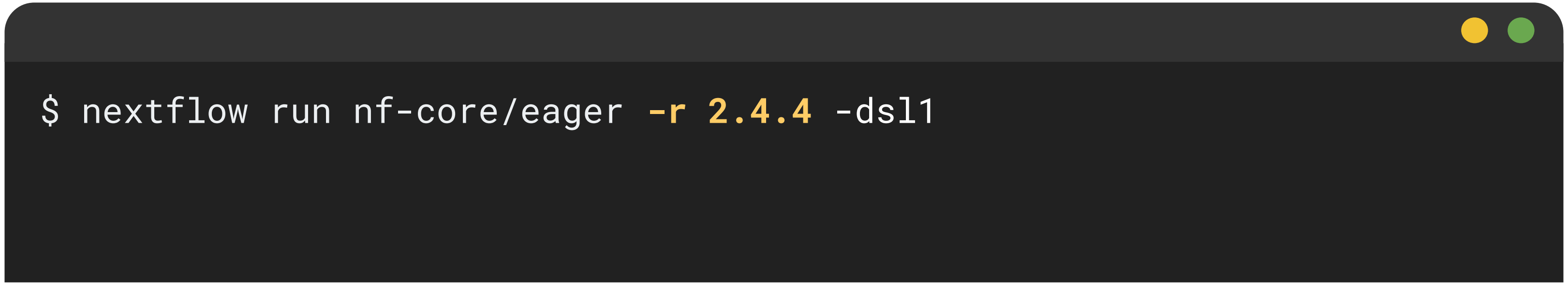
Tells nextflow to execute the EAGER pipeline

```
$ nextflow run nf-core/eager
```

Go run nf-core/eager!

🛑 Don’t press enter until we make it through the whole command!

Specify which **pipeline version** for to run for **reproducibility**



```
$ nextflow run nf-core/eager -r 2.4.4 -dsl1
```

Note: we have to currently specify `-dsl1` to specify which version of Nextflow to use. This will not be necessary in the next version of eager

Go run nf-core/eager!

🛑 Don't press enter until we make it through the whole command!

Profiles configure your analysis for specific computing environments and/or specify analytical options

```
$ nextflow run nf-core/eager -r 2.4.4 -dsl1 -profile conda
```

Go run nf-core/eager!

Don’t press enter until we make it through the whole command!

Specify a **reference** in fasta format

```
$ nextflow run nf-core/eager -r 2.4.4 -dsl1 -profile conda --fasta
../reference/GCF_001293415.1_ASM129341v1_genomic.fna
```

Go run nf-core/eager!

Don’t press enter until we make it through the whole command!

Specify input in **tsv format** or using **wildcards** (more on that later!)

```
$ nextflow run nf-core/eager -r 2.4.4 -dsl1 -profile conda --fasta
../reference/GCF_001293415.1_ASM129341v1_genomic.fna --input
ancientMetagenomeDir_eager_input_update.tsv
```

Go run nf-core/eager!

Don't press enter until we make it through the whole command!

Filter **unmapped reads** and save in fastq format

```
$ nextflow run nf-core/eager -r 2.4.4 -dsl1 -profile conda --fasta
../reference/GCF_001293415.1_ASM129341v1_genomic.fna --input
ancientMetagenomeDir_eager_input_update.tsv --run_bam_filtering
--bam_unmapped_type fastq
```

Go run nf-core/eager!

Don't press enter until we make it through the whole command!

Run genotyping using the GATK UnifiedGenotyper

```
$ nextflow run nf-core/eager -r 2.4.4 -dsl1 -profile conda --fasta
../reference/GCF_001293415.1_ASM129341v1_genomic.fna --input
ancientMetagenomeDir_eager_input_update.tsv --run_bam_filtering
--bam_unmapped_type fastq --run_genotyping --genotyping_tool ug
--gatk_ug_out_mode EMIT_ALL_SITES
```

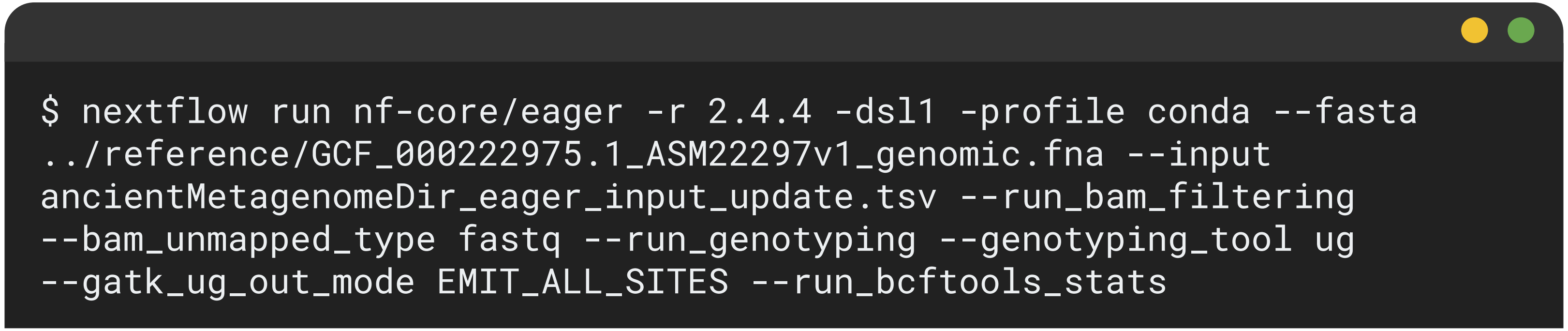
Go run nf-core/eager!

STOP Don't press enter until we make it through the whole command!

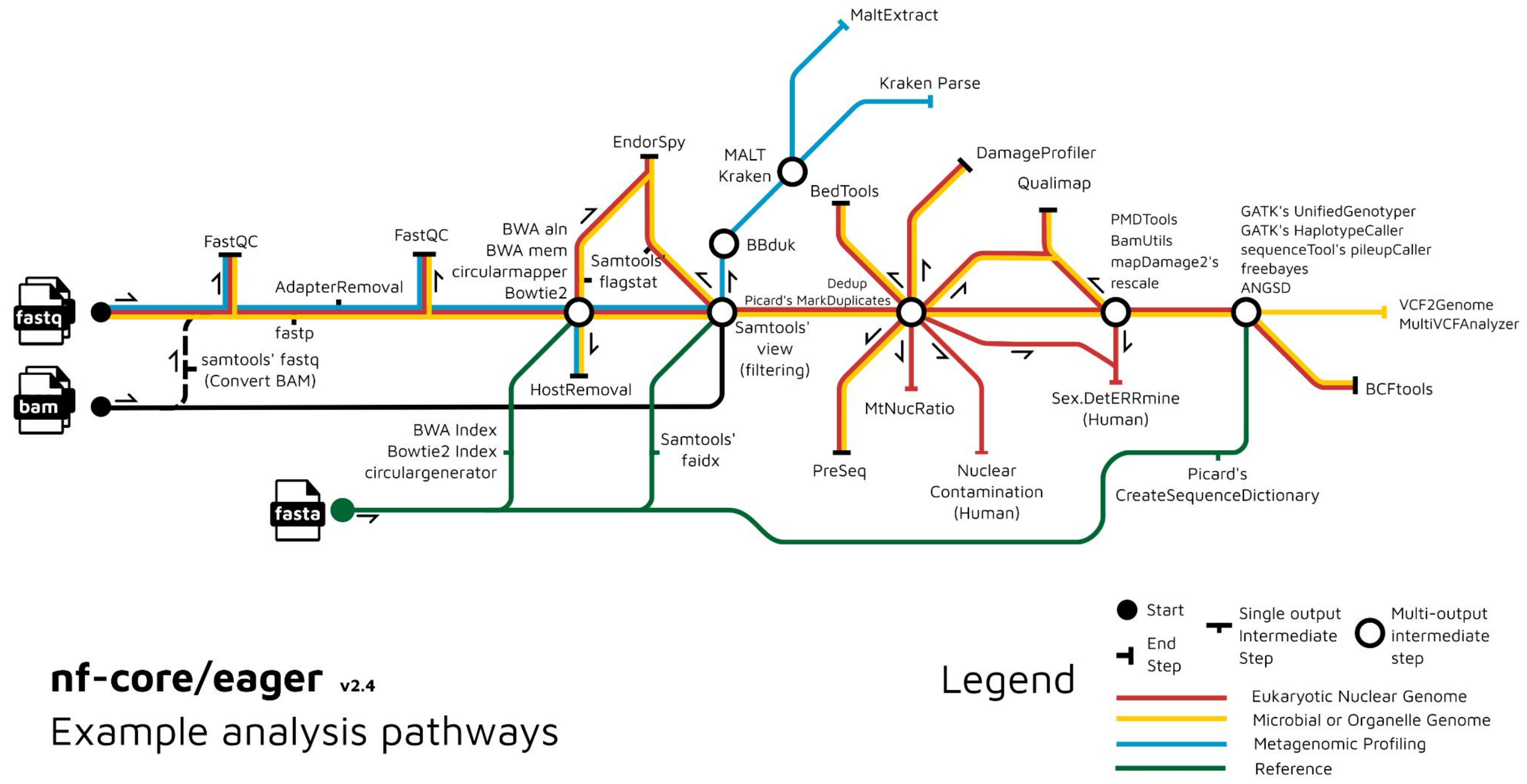
Generate **variant calling statistics**

```
$ nextflow run nf-core/eager -r 2.4.4 -dsl1 -profile conda --fasta
../reference/GCF_001293415.1_ASM129341v1_genomic.fna --input
ancientMetagenomeDir_eager_input_update.tsv --run_bam_filtering
--bam_unmapped_type fastq --run_genotyping --genotyping_tool ug
--gatk_ug_out_mode EMIT_ALL_SITES --run_bcftools_stats
```

Go run nf-core/eager!



```
$ nextflow run nf-core/eager -r 2.4.4 -dsl1 -profile conda --fasta
../reference/GCF_000222975.1_ASM22297v1_genomic.fna --input
ancientMetagenomeDir_eager_input_update.tsv --run_bam_filtering
--bam_unmapped_type fastq --run_genotyping --genotyping_tool ug
--gatk_ug_out_mode EMIT_ALL_SITES --run_bcftools_stats
```



nf-core/eager v2.4

Example analysis pathways

And now we wait…

Other options: A teaser

Skip steps (e.g. for preprocessed, adapter-trimmed data)

```
$ nextflow run nf-core/eager -r 2.4.4 -dsl1 -profile conda --fasta
‘../reference/GCF_001293415.1_ASM129341v1_genomic.fna’ --input
‘ancientMetagenomeDir_eager_input.tsv’ --run_bam_filtering
--bam_unmapped_type 'fastq' --run_genotyping --genotyping_tool 'ug'
--gatk_ug_out_mode EMIT_ALL_SITES --run_bcftools_stats --skip_fastqc
--skip_adapterremoval
```

⚠️ For full parameter documentation, see https://nf-co.re/eager/2.4.4/parameters

Other options: A teaser

Trim bases from fastq files (e.g. to remove damage from UDG-half treated libraries)

```
$ nextflow run nf-core/eager -r 2.4.4 -dsl1 -profile conda --fasta
‘../reference/GCF_001293415.1_ASM129341v1_genomic.fna’ --input
‘ancientMetagenomeDir_eager_input.tsv’ --run_bam_filtering
--bam_unmapped_type 'fastq' --run_genotyping --genotyping_tool 'ug'
--gatk_ug_out_mode EMIT_ALL_SITES --run_bcftools_stats
--run_post_ar_trimming --post_ar_trim_front 2 --post_ar_trim_tail 2
--post_ar_trim_front2 2 --post_ar_trim_tail2 2
```

⚠️ For full parameter documentation, see https://nf-co.re/eager/2.4.4/parameters

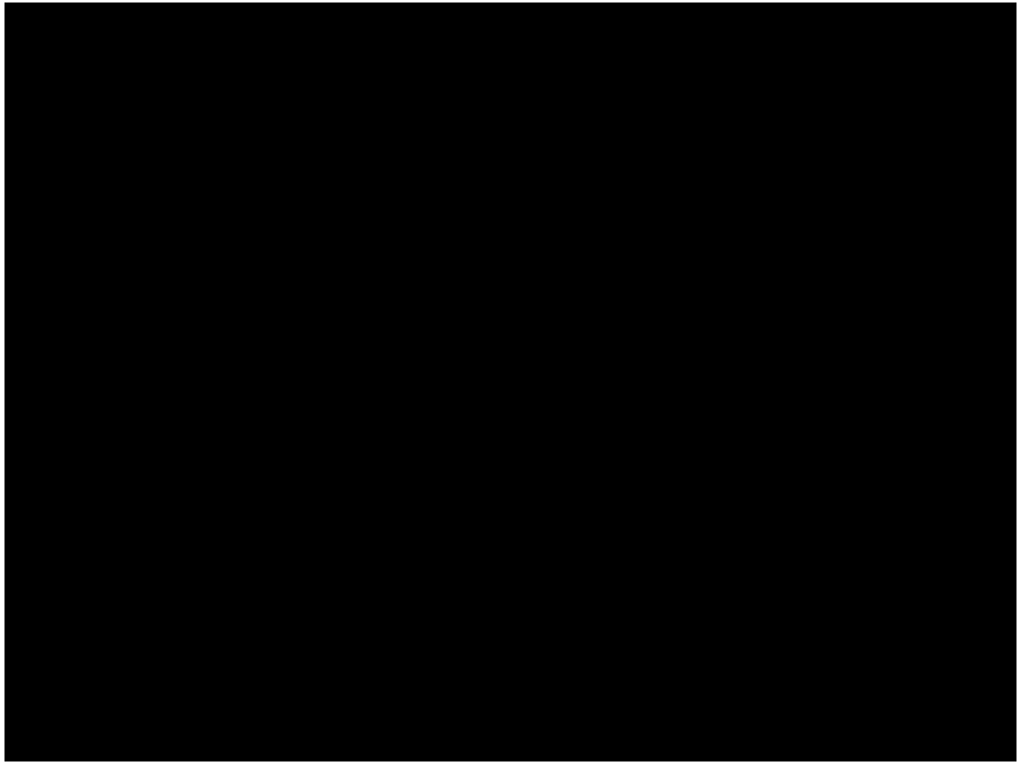
Other options: A teaser

Adjust mapping parameters (e.g. for single-stranded libraries with lots of damage)

```
$ nextflow run nf-core/eager -r 2.4.4 -dsl1 -profile conda --fasta
'../reference/GCF_001293415.1_ASM129341v1_genomic.fna' --input
'ancientMetagenomeDir_eager_input.tsv' --run_bam_filtering
--bam_unmapped_type 'fastq' --run_genotyping --genotyping_tool 'ug'
--gatk_ug_out_mode EMIT_ALL_SITES --run_bcftools_stats --bwaalnn 0.01
--bwaalnl 16
```

⚠️ For full parameter documentation, see https://nf-co.re/eager/2.4.4/parameters

And much, much more!

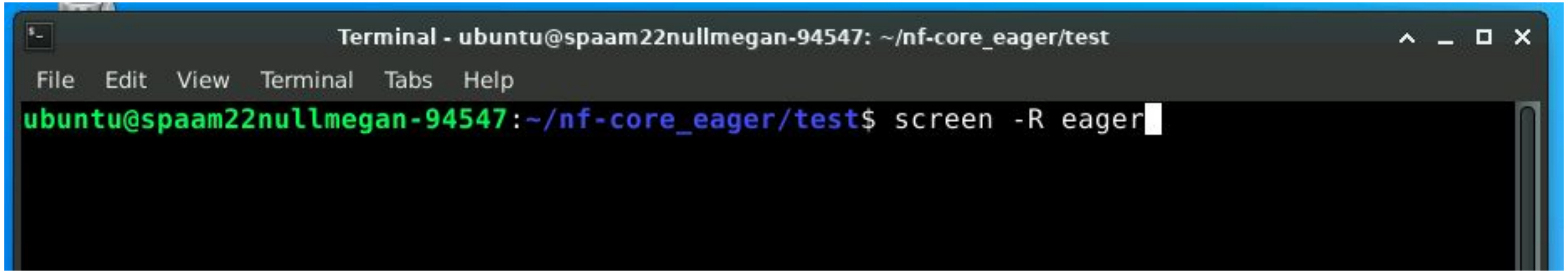


Top tips for EAGER success

While our commands are running…

Tip 1: Screen sessions

- Depending on input data, infrastructure, and desired analysis, running nf-core/eager can take **hours or even days**
- To avoid crashes due to loss of power or network connectivity, try running nf-core/eager in a **screen** or **tmux** session


Terminal - ubuntu@spaam22nullmegan-94547: ~/nf-core_eager/test
File Edit View Terminal Tabs Help

```
ubuntu@spaam22nullmegan-94547:~/nf-core_eager/test$ screen -R eager
```

Tip 2: Multiple ways to supply input data

1. **Wildcards-** useful for ‘fast’ and simple input data
   a. Same sequencing instrument/configuration
   b. Stored in a common location
   c. One file per sample

```
$ nextflow run nf-core/eager -r 2.4.4 -dsl1 -profile conda --fasta
../reference/GCF_001293415.1_ASM129341v1_genomic.fna --input
../data/*fastq.gz
```

Tip 2: Multiple ways to supply input data

2. **Input tsv files-** more powerful!!
    a. Supplies nf-core/eager with details on input data
    b. nf-core/eager can ‘intelligently’ apply analyses to certain files only (e.g. merging for paired end sequencing, poly-G trimming for NextSeq data)
    c. Efficient merging of output by library/sample -> useful where there are multiple samples per individual

Tip 2: Multiple ways to supply input data

2. **Input tsv files-** more flexible!!

What does our tsv input look like?

```
$ cat ancientMetagenomeDir_eager_input.tsv
```

Tip 3: Get your report via email

- If your HPC has GNU mail or sendmail setup, get your MultiQC html report delivered via email!
- Also serves as an alert that your run is finally finished 😎

```
$ nextflow run nf-core/eager -r 2.4.4 -dsl1 -profile conda --fasta
'../reference/GCF_001293415.1_ASM129341v1_genomic.fna' --input
'ancientMetagenomeDir_eager_input.tsv' --run_bam_filtering
--bam_unmapped_type 'fastq' --run_genotyping --genotyping_tool 'ug'
--gatk_ug_out_mode EMIT_ALL_SITES --run_bcftools_stats --email
'megan_michel@eva.mpg.de'
```

Tip 4: Check out the EAGER GUI

Go to https://nf-co.re/eager/launch

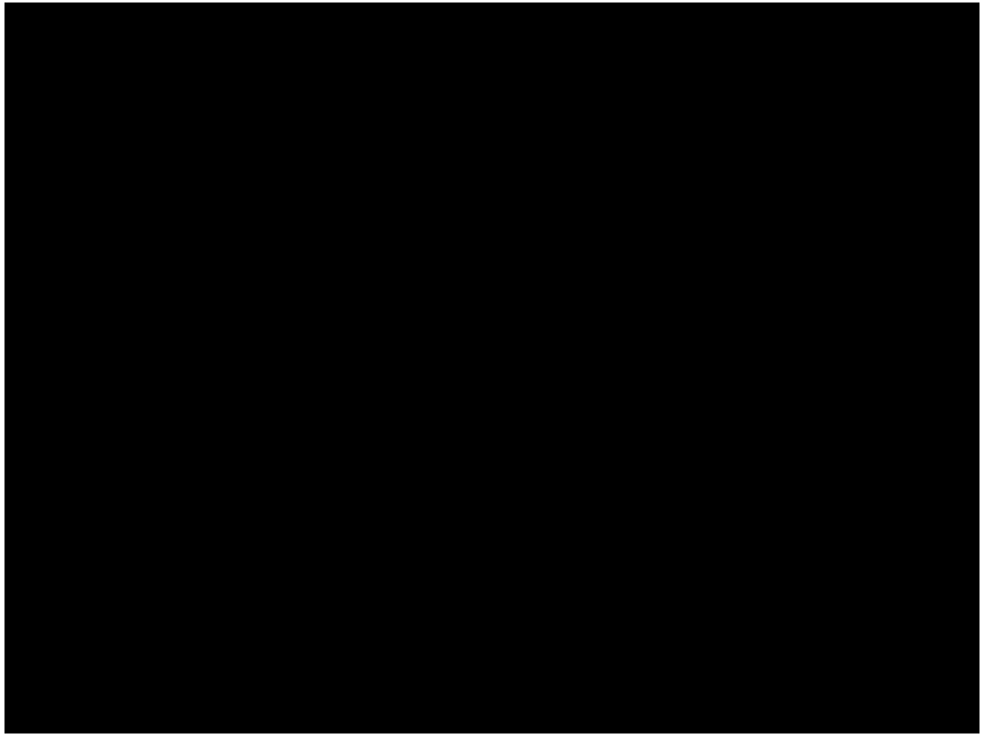
Tip 5: When something fails… all is not lost!

- When an **individual job fails**…
  - nf-core/eager will **automatically resubmit that job*** with double the memory and CPUs
  - Can occur up to two times per job
- When the **pipeline crashes**…
  - Resubmit with **-resume**
  - Nextflow can **retrieved cached results** from previous steps, as long as the input is the same
  - Saves time and computational resources!

* in most cases

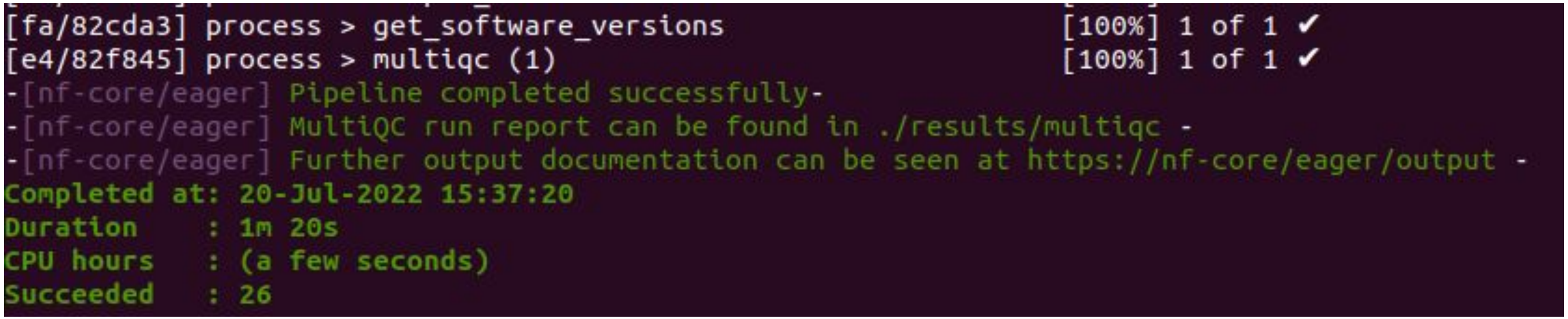
Tip 6: Nextflow Tower (for regular users)

More information here: https://help.tower.nf/22.2/



Output

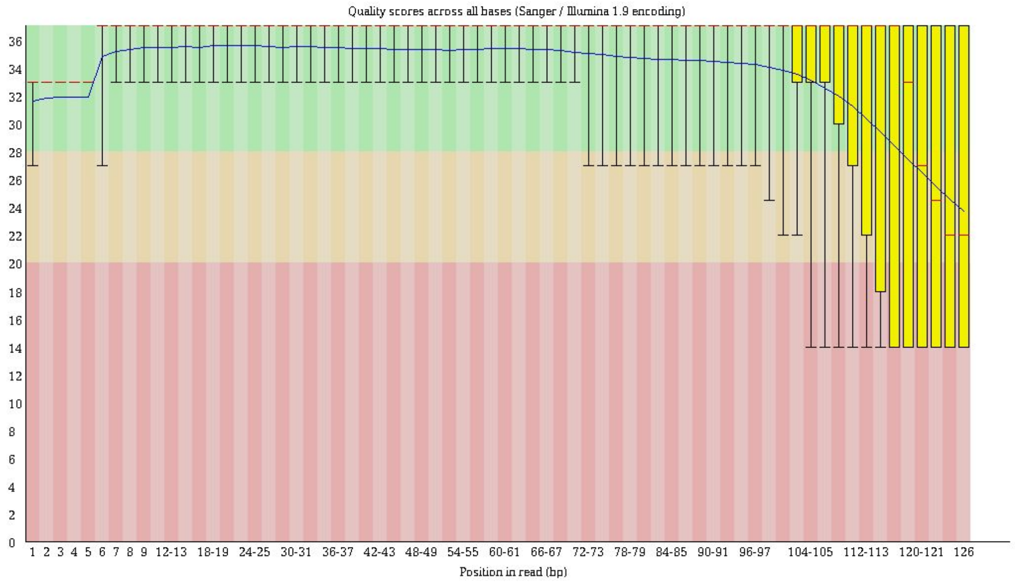
🎉🎉🎉🎉🎉🎉🎉🎉🎉🎉🎉🎉🎉🎉🎉🎉🎉🎉



```
[fa/82cda3] process > get_software_versions                          [100%] 1 of 1 ✔
[e4/82f845] process > multiqc (1)                                    [100%] 1 of 1 ✔
-[nf-core/eager] Pipeline completed successfully-
-[nf-core/eager] MultiQC run report can be found in ./results/multiqc -
-[nf-core/eager] Further output documentation can be seen at https://nf-core/eager/output -
Completed at: 20-Jul-2022 15:37:20
Duration     : 1m 20s
CPU hours    : (a few seconds)
Succeeded    : 26
```

Overview

- Common file outputs for downstream
- How to ‘quality control’ your run via MultiQC

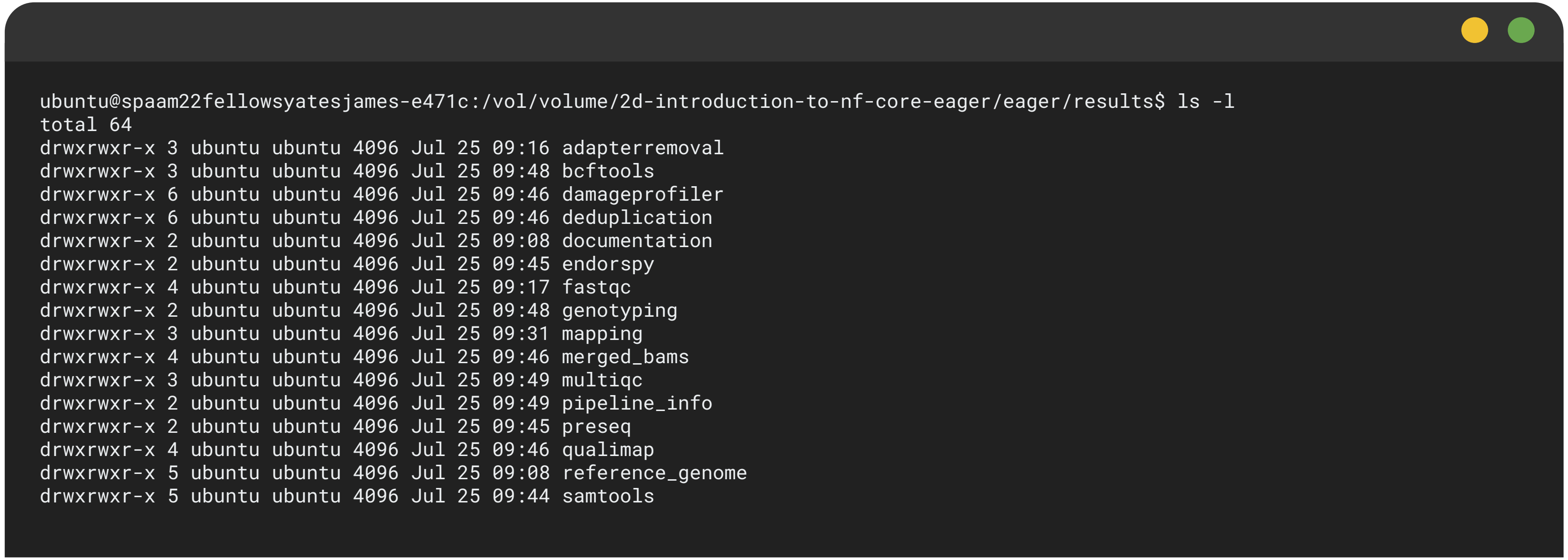


Output files

Which output files to look for - depends on context!

- Genome reconstruction:
  - FASTQ: for downstream re-mapping, re-profiling, upload
  - BAM (de-duplicated): manual inspection, variant calling, damage patterns
  - VCF: for variant calls/genotypes
  - FASTA : for multi-sequence alignment, phylogenetic analysis
- Microbiome reconstruction
  - TSV: for OTU tables
- General: MultiQC!

Output files



```
ubuntu@spaam22fellowsyatesjames-e471c:/vol/volume/2d-introduction-to-nf-core-eager/eager/results$ ls -l
total 64
drwxrwxr-x 3 ubuntu ubuntu 4096 Jul 25 09:16 adapterremoval
drwxrwxr-x 3 ubuntu ubuntu 4096 Jul 25 09:48 bcftools
drwxrwxr-x 6 ubuntu ubuntu 4096 Jul 25 09:46 damageprofiler
drwxrwxr-x 6 ubuntu ubuntu 4096 Jul 25 09:46 deduplication
drwxrwxr-x 2 ubuntu ubuntu 4096 Jul 25 09:08 documentation
drwxrwxr-x 2 ubuntu ubuntu 4096 Jul 25 09:45 endorspy
drwxrwxr-x 4 ubuntu ubuntu 4096 Jul 25 09:17 fastqc
drwxrwxr-x 2 ubuntu ubuntu 4096 Jul 25 09:48 genotyping
drwxrwxr-x 3 ubuntu ubuntu 4096 Jul 25 09:31 mapping
drwxrwxr-x 4 ubuntu ubuntu 4096 Jul 25 09:46 merged_bams
drwxrwxr-x 3 ubuntu ubuntu 4096 Jul 25 09:49 multiqc
drwxrwxr-x 2 ubuntu ubuntu 4096 Jul 25 09:49 pipeline_info
drwxrwxr-x 2 ubuntu ubuntu 4096 Jul 25 09:45 preseq
drwxrwxr-x 4 ubuntu ubuntu 4096 Jul 25 09:46 qualimap
drwxrwxr-x 5 ubuntu ubuntu 4096 Jul 25 09:08 reference_genome
drwxrwxr-x 5 ubuntu ubuntu 4096 Jul 25 09:44 samtools
```

But before…

Let’s check everything looks ‘normal’*

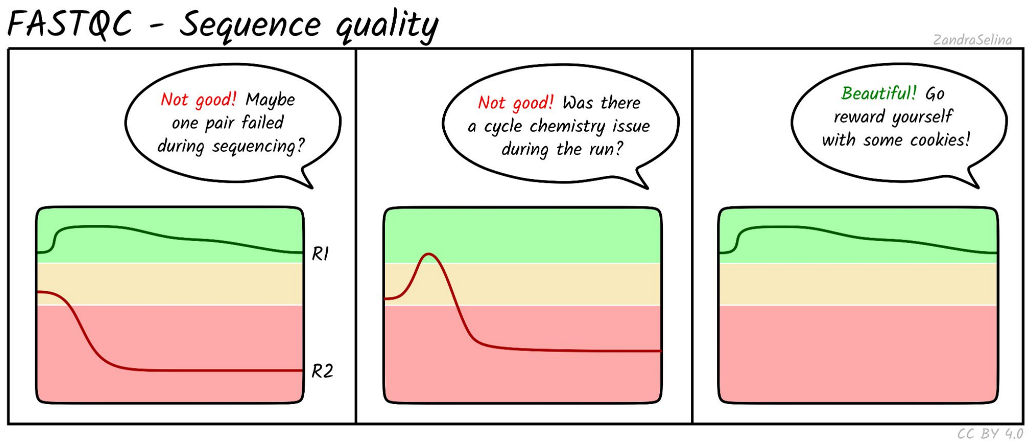
- Did sequencing run go well (moneys worth/sufficient data)?
- Any artefacts (remaining adapters? over-amplification?)
- Sufficient coverage?
- Actually have damage?
- Can we sequence more?
- Contamination?

* takes experience and practise! Often via ‘intutition’

Your main friend

nf-core/eager documentation!

https://nf-co.re/eager/output


FASTQC - Sequence quality
ZandraSelina
Not good! Maybe one pair failed during sequencing?
R1
R2
Not good! Was there a cycle chemistry issue during the run?
Beautiful! Go reward yourself with some cookies!
CC BY 4.0

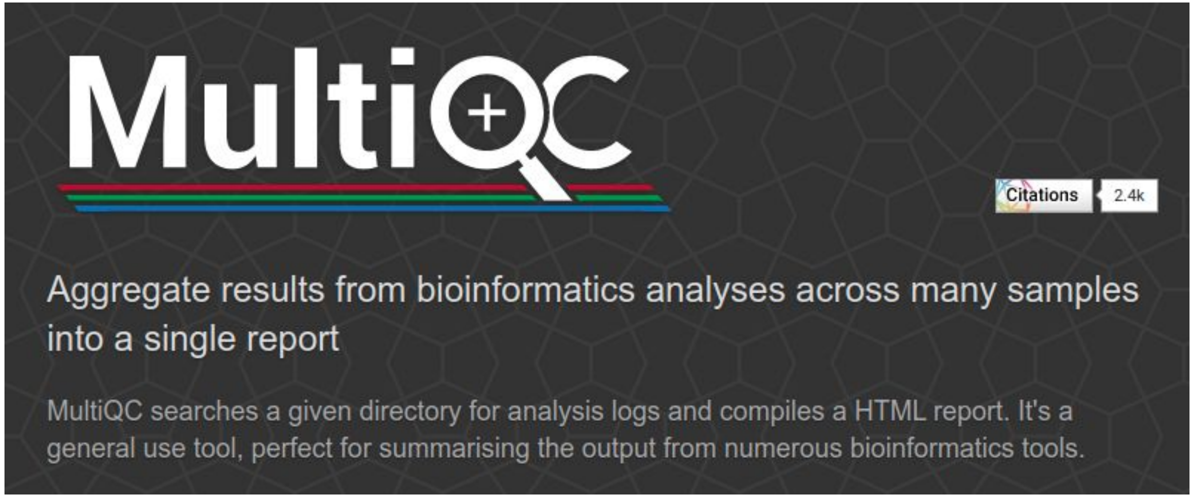
Copy also in output directory of every run!

Note: Text MIT & Images CC-BY 4.0 → reuse for your own training material (with attribution 😉)

MultiQC

What is MultiQC?

Witchcraft!



Aggregate results from bioinformatics analyses across many samples into a single report

MultiQC searches a given directory for analysis logs and compiles a HTML report. It's a general use tool, perfect for summarising the output from numerous bioinformatics tools.

Use for your own projects! `multiqc <your_directory>/`

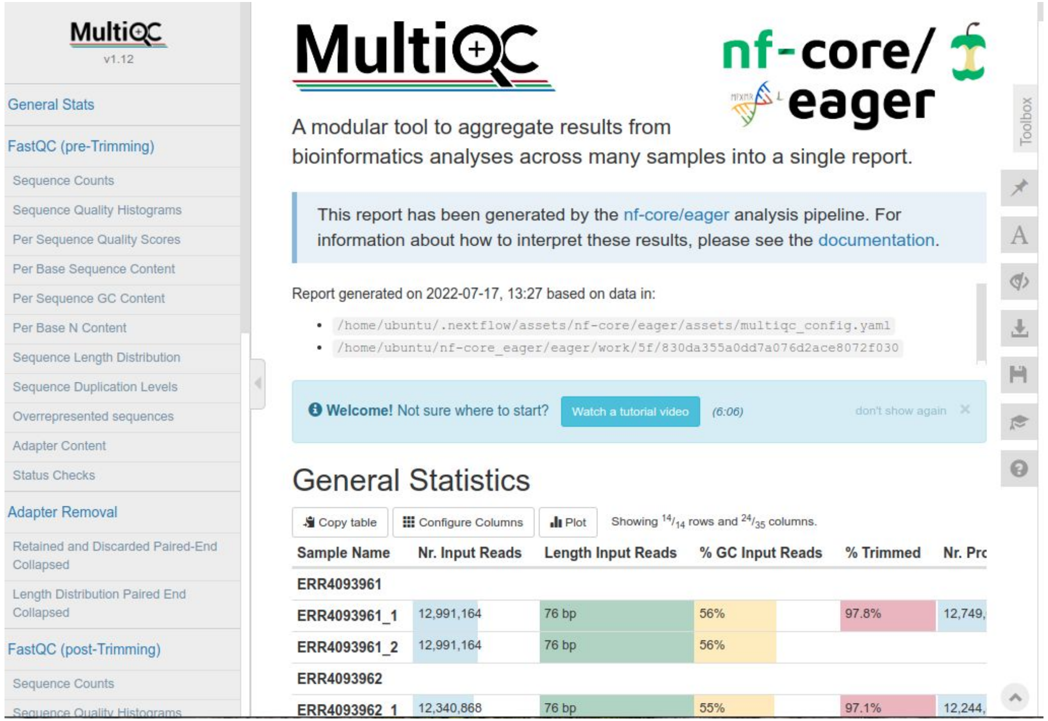
Open your MultiQC report

Either go to your file browser, navigate to eager results and double click the `multiqc_report.html` file (should open your web browser)

```
$ cd /vol/volume/2d-introduction-to-nf-core-eager/
$ firefox multiqc_report.html
```

Note: if mail or sendmail is set up on your server/HPC: `--email <your>@<email>.com` to your nextflow command may see your report in your inbox at the end of the run 😎

First Impressions?



MultiQC v1.12

- General Stats
- FastQC (pre-Trimming)
  - Sequence Counts
  - Sequence Quality Histograms
  - Per Sequence Quality Scores
  - Per Base Sequence Content
  - Per Sequence GC Content
  - Per Base N Content
  - Sequence Length Distribution
  - Sequence Duplication Levels
  - Overrepresented sequences
  - Adapter Content
  - Status Checks
- Adapter Removal
  - Retained and Discarded Paired-End Collapsed
  - Length Distribution Paired End Collapsed
- FastQC (post-Trimming)
  - Sequence Counts
  - Sequence Quality Histograms

MultiQC nf-core/eager

A modular tool to aggregate results from bioinformatics analyses across many samples into a single report.

This report has been generated by the nf-core/eager analysis pipeline. For information about how to interpret these results, please see the documentation.

Report generated on 2022-07-17, 13:27 based on data in:

- `/home/ubuntu/.nextflow/assets/nf-core/eager/assets/multiqc_config.yaml`
- `/home/ubuntu/nf-core_eager/eager/work/5f/830da355a0dd7a076d2ace8072f030`

Welcome! Not sure where to start? Watch a tutorial video (6:06) don't show again

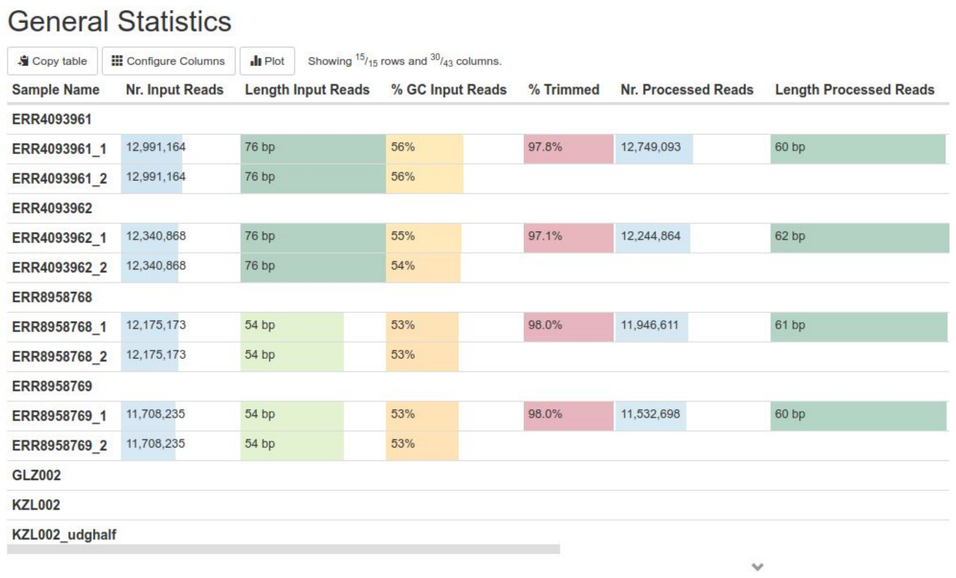
General Statistics

Copy table | Configure Columns | Plot | Showing $^{14}/_{14}$ rows and $^{24}/_{35}$ columns.

Sample Name	Nr. Input Reads	Length Input Reads	% GC Input Reads	% Trimmed	Nr. Pro
ERR4093961					
ERR4093961_1	12,991,164	76 bp	56%	97.8%	12,749,
ERR4093961_2	12,991,164	76 bp	56%		
ERR4093962					
ERR4093962_1	12,340,868	76 bp	55%	97.1%	12,244,

Click, hover, filter, export, create your own plots!

General Stats



General Statistics

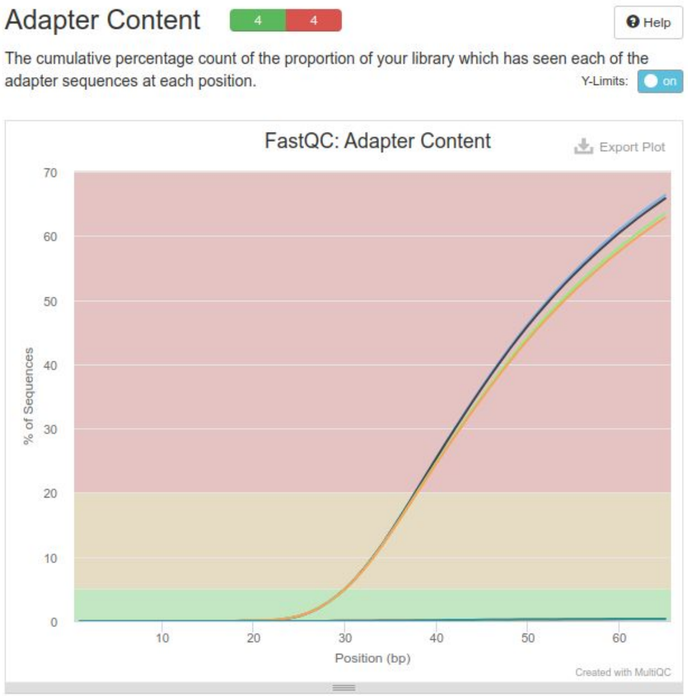
Copy table | Configure Columns | Plot | Showing 15/15 rows and 30/43 columns.

Sample Name	Nr. Input Reads	Length Input Reads	% GC Input Reads	% Trimmed	Nr. Processed Reads	Length Processed Reads
ERR4093961						
ERR4093961_1	12,991,164	76 bp	56%	97.8%	12,749,093	60 bp
ERR4093961_2	12,991,164	76 bp	56%			
ERR4093962						
ERR4093962_1	12,340,868	76 bp	55%	97.1%	12,244,864	62 bp
ERR4093962_2	12,340,868	76 bp	54%			
ERR8958768						
ERR8958768_1	12,175,173	54 bp	53%	98.0%	11,946,611	61 bp
ERR8958768_2	12,175,173	54 bp	53%			
ERR8958769						
ERR8958769_1	11,708,235	54 bp	53%	98.0%	11,532,698	60 bp
ERR8958769_2	11,708,235	54 bp	53%			
GLZ002						
KZL002						
KZL002_udghalf						

Things to look out for

- Multiple lines per sample!
- General overview
- Look for expected reads
- Look for outliner numbers
- Configure columns to collapse rows per library/sample
  - Multiple lines per sample if: multiple lanes, libraries etc until 'merged' step of pipeline
- Play with custom 'Plots'

Sequencing QC: FastQC


Adapter Content

The cumulative percentage count of the proportion of your library which has seen each of the adapter sequences at each position.

Things to look out for

- ✅ Expected number of reads?
- ✅ Short sequence lengths
  - Early indicator of fragmented aDNA!
- ✅ Large fraction of adapter in reads
  - Early indicator of fragmented aDNA
- ⚠️ High amount of duplicates
- ⚠️ Large or 'early' amount of 'red' (low quality bases) in reads
- ⚠️ Outlier libraries?

Sequencing QC: AdapterRemoval

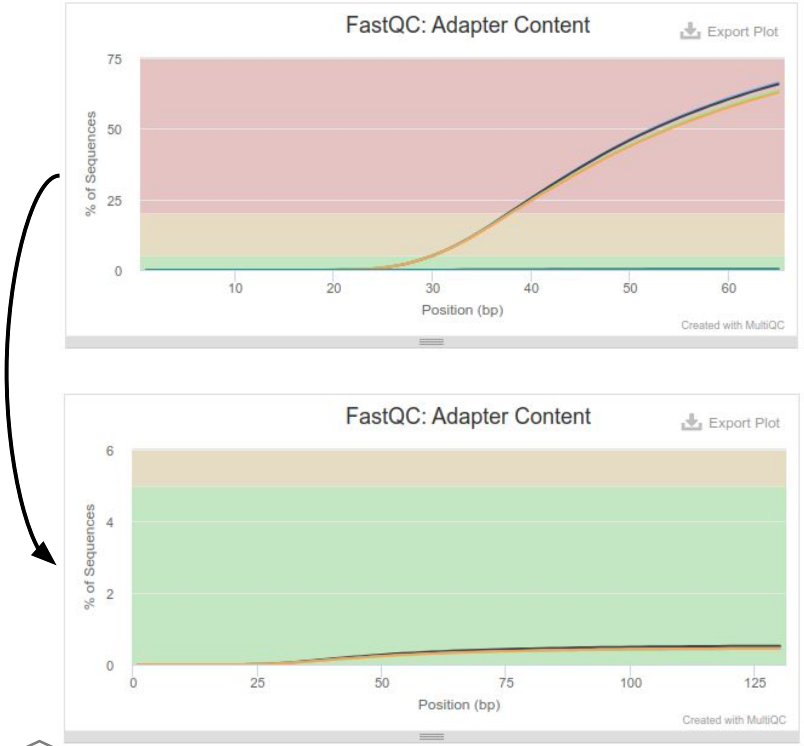
Nothing to see here 👀*

Things to look out for

- ✅ Large fraction of reads collapsed
  - Early indicator of fragmented aDNA
- ⚠️ Large numbers of discarded reads
- ⚠️ Peak of read length plot >75bp

*(*Plot in latest version of MultiQC version a mess - fixed in upcoming 1.13)*

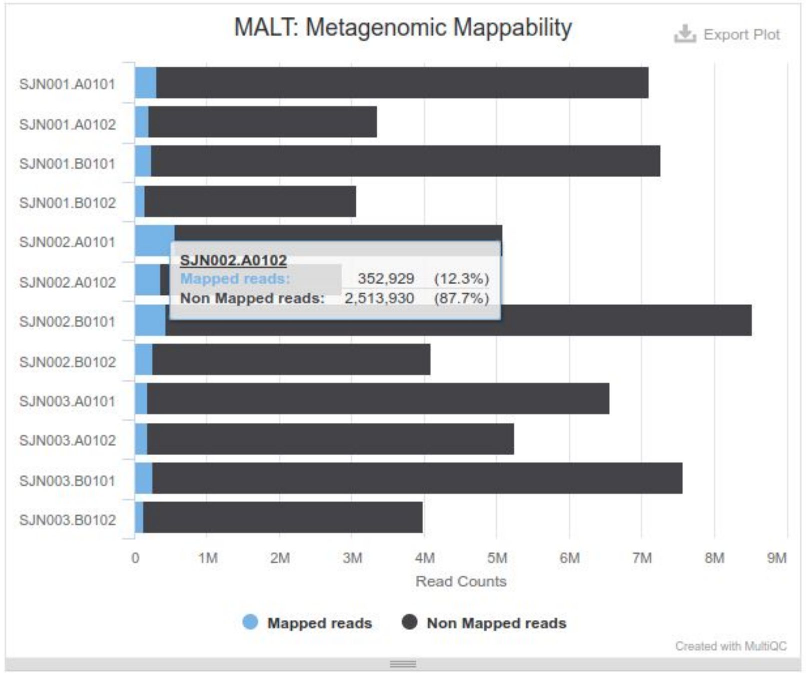
FastQC (post-trimming)



Things to look out for

- ✅ More 'green' in plots than pre-Adapter Removal
- ⚠️ No remaining artefacts

Metagenomic Classification (MALT/Kraken)

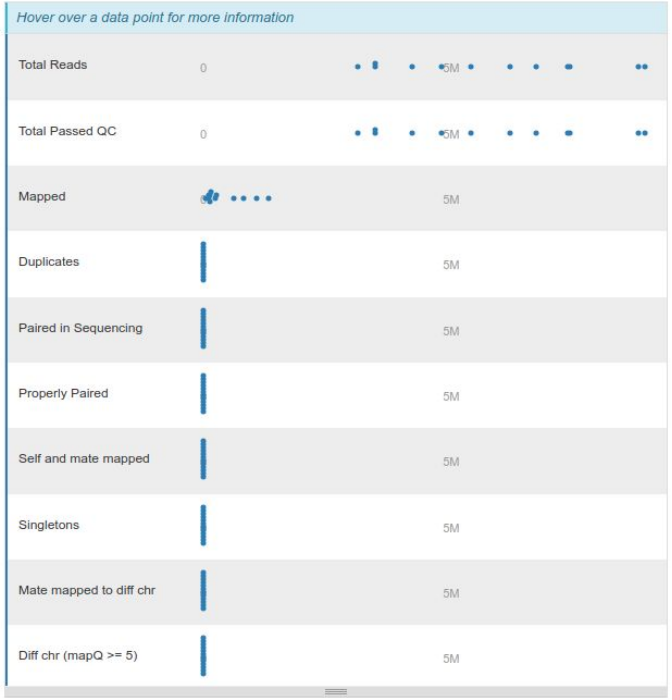


SJN002.A0102
Mapped reads: 352,929 (12.3%)
Non Mapped reads: 2,513,930 (87.7%)

Things to look out for

- ✅ High mappability - ideal!
- ✅ Low mappability - sort of expected!
  - Database bias, lots of uncharacterised environmental taxa
- ⚠️ Low no. taxonomically assigned reads

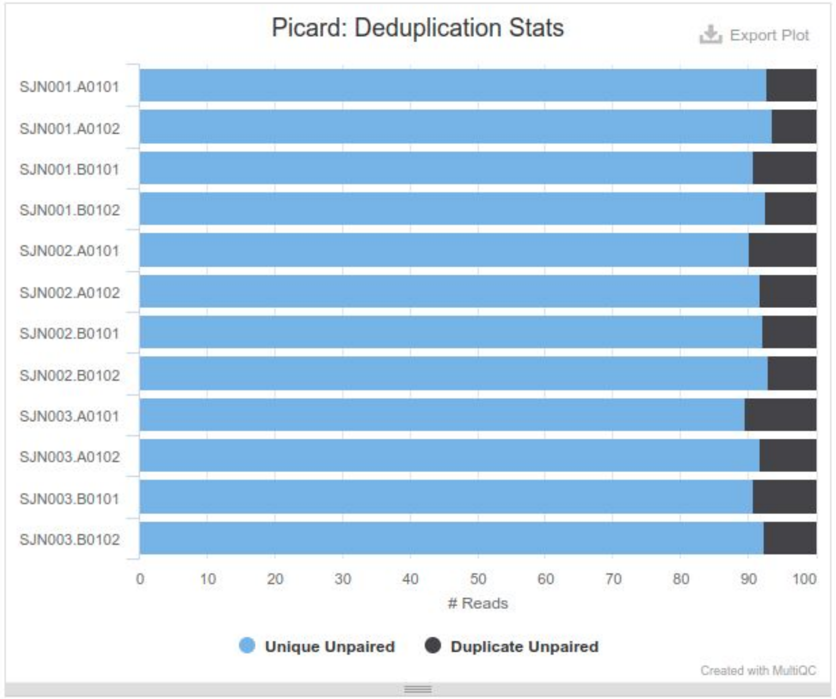
Mapping QC: flagstat



Things to look out for

- ✅ High numbers of mapped reads
- ⚠️ Clear outliers in number of mapped reads

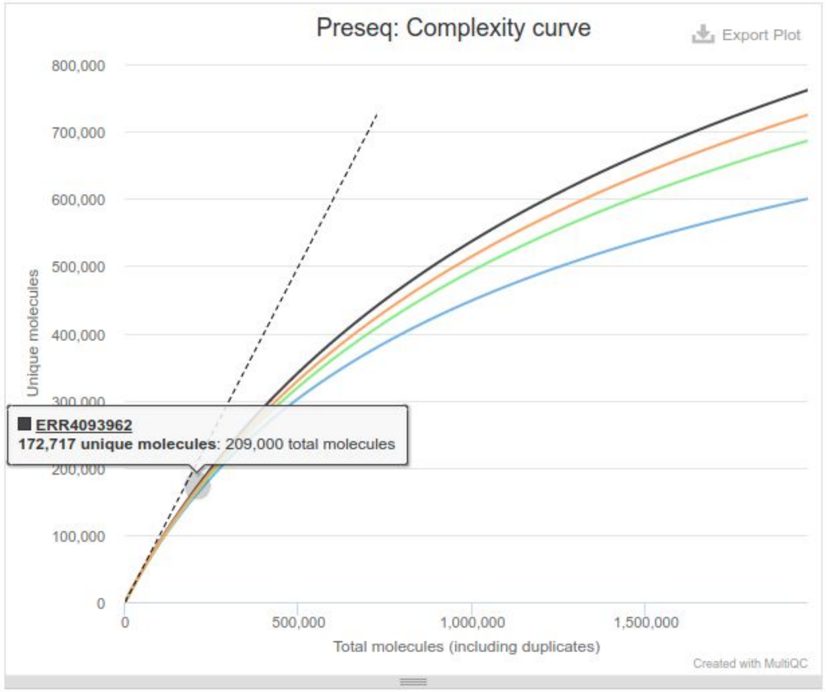
Library QC: Mark Duplicates



Things to look out for

- ✅ High numbers of unique reads
- ⚠️ High numbers of duplicates
  - Note: differences between capture/shotgun data

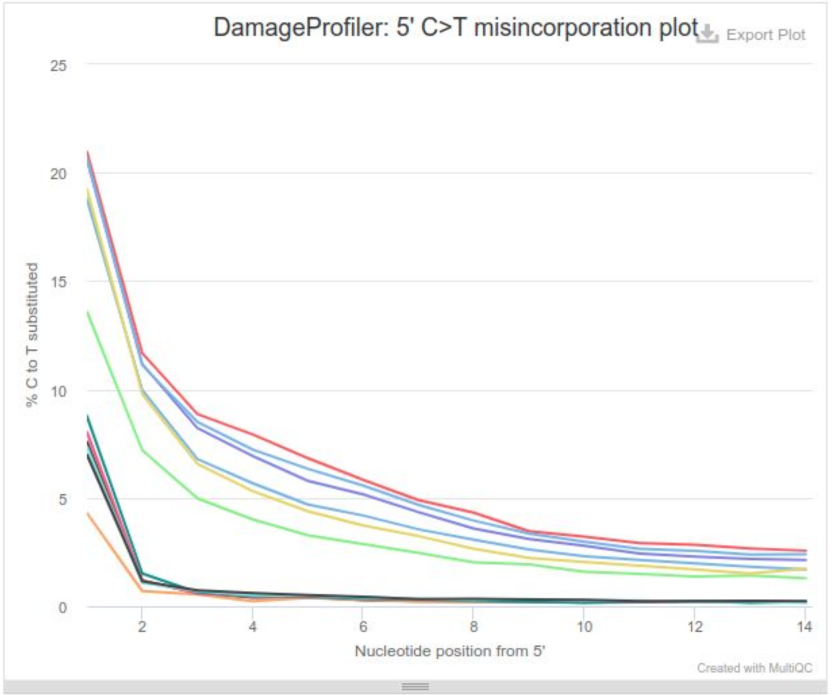
Library QC: preseq



Things to look out for

- ✅ Follow dashed (theoretical) line
  - 1:1 ratio unique : total molecules
- ⚠️ Early plateauing

Palaeogenomic QC: DamageProfiler

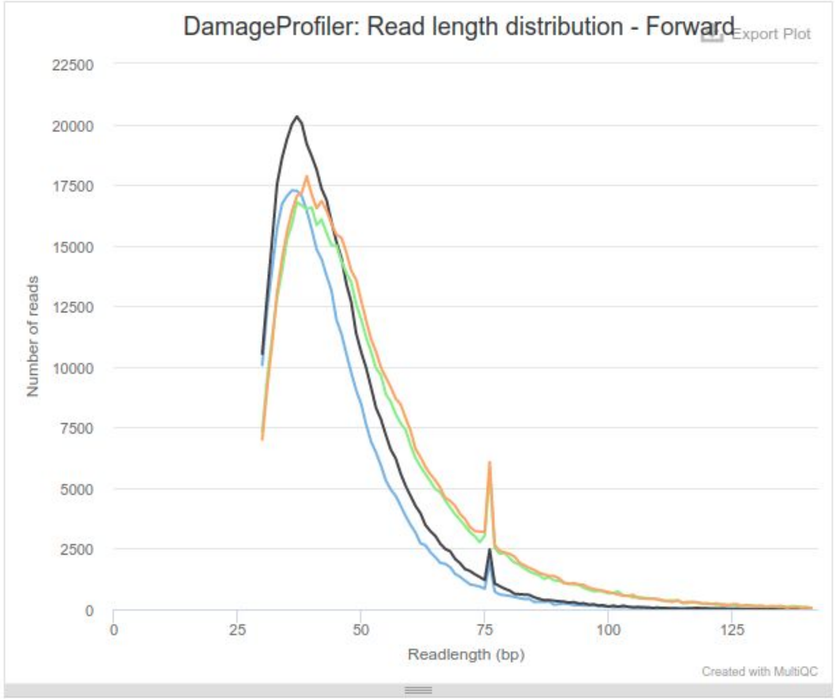


Things to look out for

- ✅ Decreasing curve base 0bp → inside read
    - Note: differences in UDG treatment!

      UDG full: no damage, partial: 1/2bp only
- ✅ High % values Y axis
- ⚠️ Noisey/bouncy curves
- ⚠️ High frequency across entire read
- ⚠️ Random peaks in middle of read
- ⚠️ Low frequency of C > T / G > A
    - Note: differences between single- and double-stranded libraries!

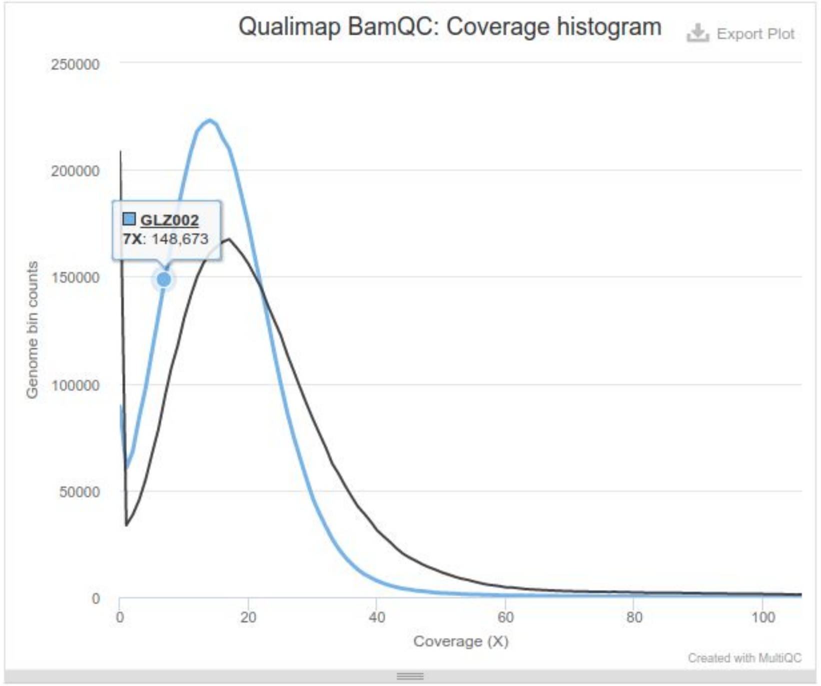
Palaeogenomic QC: DamageProfiler



Things to look out for

- ✅ Right skew distribution
- ✅ Peak low values on x-axis
    - But not too low! <25bp difficult!
- ⚠️ Peak at high values on x-axis
- ⚠️ Multiple (clear) peaks

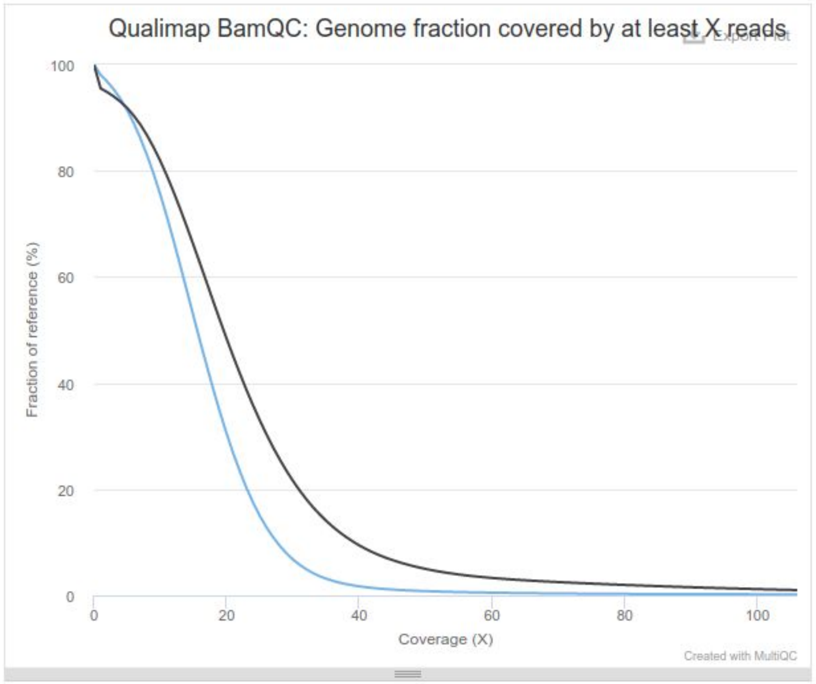
(Palaeo)genomic QC: qualimap



Things to look out for

- ✅ Single peak at high X Coverage
  - >5 pretty good for aDNA
- ⚠️ Multiple peaks/unsmooth curve
- ⚠️ Peak at very low values
-

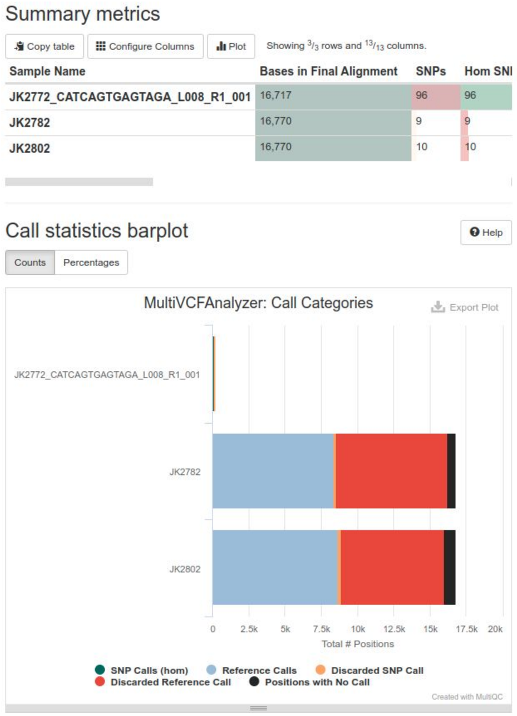
(Palaeo)genomic QC: qualimap



Things to look out for

- ✅ Large ‘area under the curve’
- ✅ High X coverage at high % of reference
- ⚠️ Low % covered

Paleogenomics: (genomic) contamination



Summary metrics

Showing 3/3 rows and 13/13 columns.

Sample Name	Bases in Final Alignment	SNPs	Hom SNI
JK2772_CATCAGTGAGTAGA_L008_R1_001	16,717	96	96
JK2782	16,770	9	9
JK2802	16,770	10	10

Call statistics barplot

Things to look out for

- ✅ Some/consistent no. homozygous SNPs
- ⚠️ High number of heterozygous SNPs
  - Indicator of cross-contamination
- ⚠️ Outliers of too-high or too-low SNP values
  - Indicators of wrong/too similar reference genome
- ⚠️ High number discarded SNPs

Summary: nf-core/eager output

- Various output formats
  - FASTQ (raw reads)
  - BAM (mapped reads)
  - VCFs (variant calls)
  - FASTAs (consensus sequence)
  - TSV (various, often OTU tables)
- Usage: context dependent!
- nf-core/eager output directories
  - As much choice to user!
  - Output files and raw log files

- MultiQC plots
  - Many plots
  - Takes experience/'feeling' to rapidly evaluate
  - Mostly checking for outliers
  - Be aware of 'failure' reporting often designed for modern DNA!

Read the documentation!

https://nf-core/eager/output