

Bioinformatics

ANDREW WEBB

Overview for our next 3 lectures

Intro to bioinformatics

- Focus on whole-genome sequencing
- Discussion of other data types

Run whole-genome sequencing pipeline on Princeton's HPC

- Simple, made from scratch

Converting the above pipeline into a snakemake workflow

What is Bioinformatics?

Wikipedia

- **Bioinformatics** is an interdisciplinary field of science that develops methods and software tools for understanding biological data, especially when the data sets are large and complex.

Coined by Paulien Hogeweg & Ben Hesper

- The study of informatic processes in biotic systems

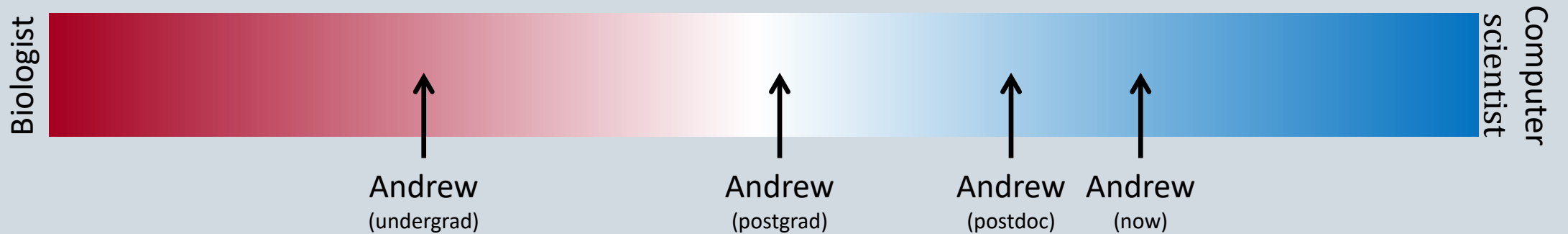
Other interpretations/opinions – i.e. a biological focused

- Data analyst
- Data wrangler
- Software developer
- Information theorist

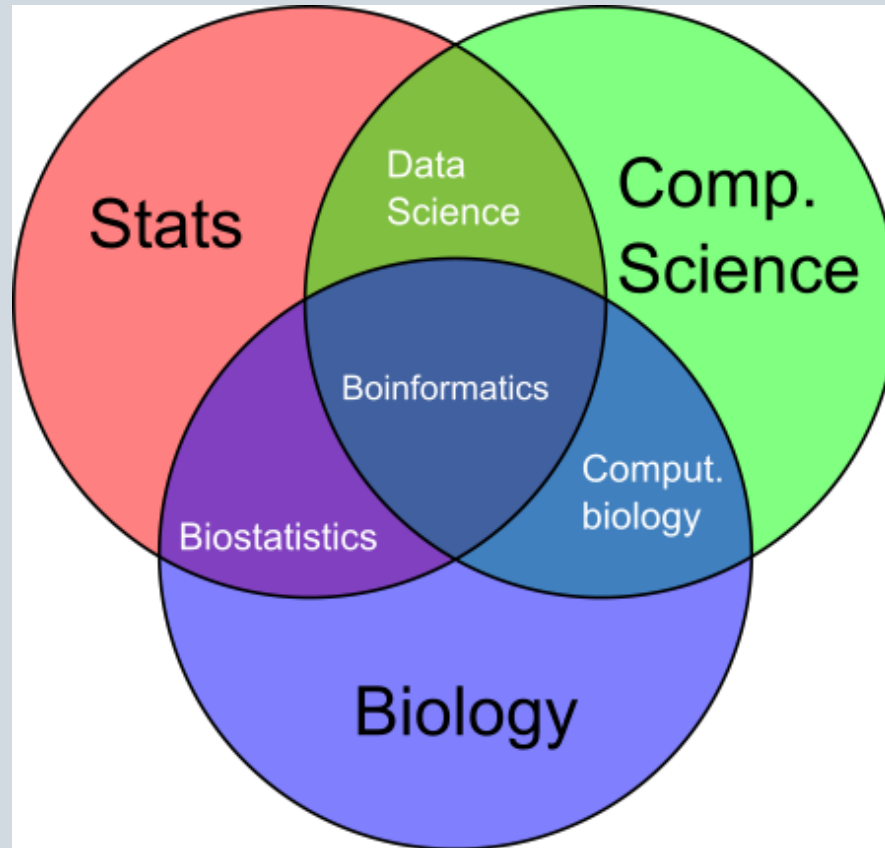
And... often synonymous with computational biologist/geneticist/genomicist

What is Bioinformatics?

All interpretations are correct



What is Bioinformatics?



What would you say...you do here?

Analysis of biological data (often large scale)

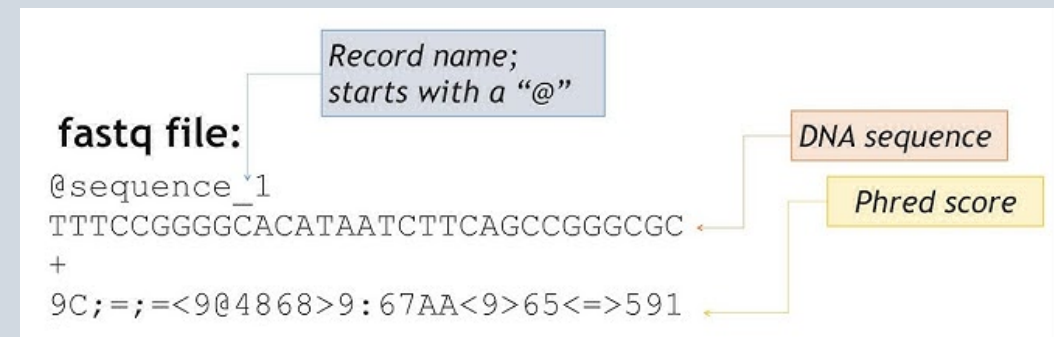
- Mostly: DNA & RNA
- Less often: Proteins & metabolites

Analyzing

- A single organism, organ, or tissue
- Pooled samples (using barcodes)
- An individual cell

Requires

- An expertise in the operation of software/packages and data formats to analyze the data
- Enough programming knowledge to write scripts for data manipulation and reporting



Basic local alignment search tool

Stephen F. Altschul¹, Warren Gish¹, Webb Miller², Eugene W. Myers³, David J. Lipman¹

[Show more](#) ▼

Cited by 114835

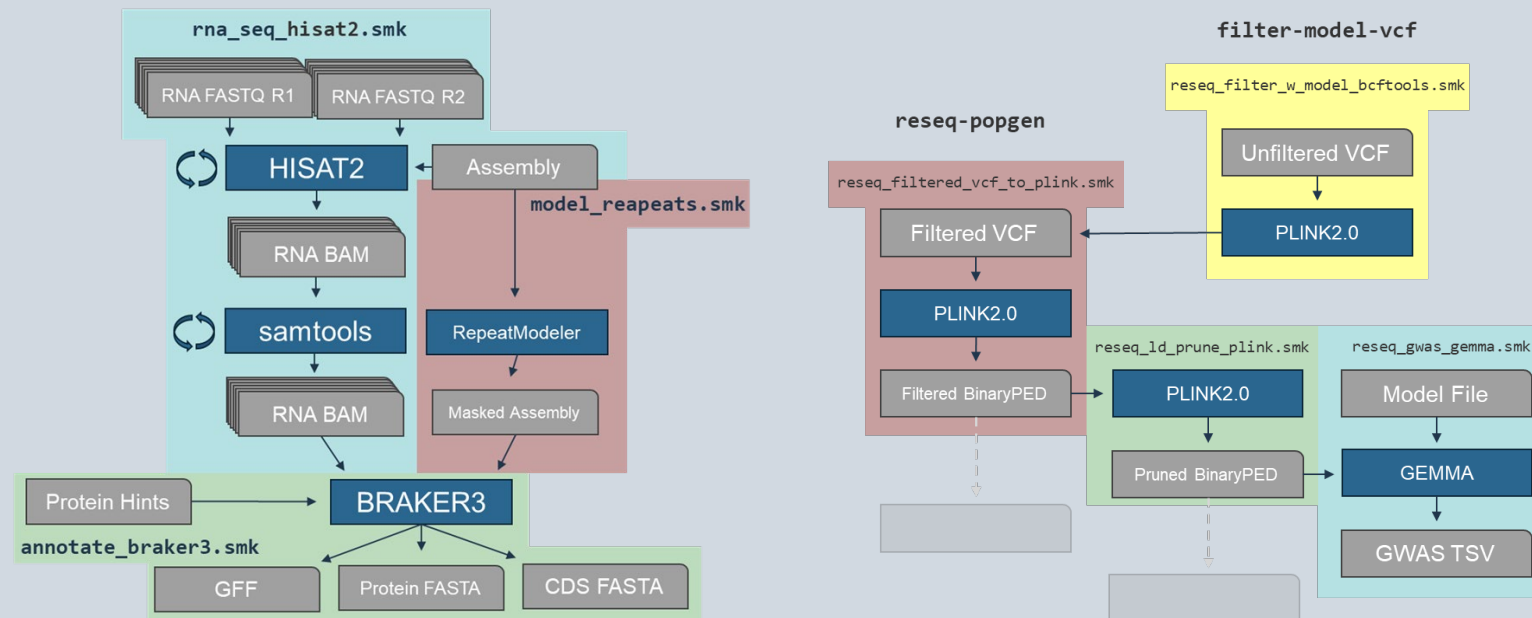
[+ Add to Mendeley](#) [Share](#) [Cite](#)

[https://doi.org/10.1016/S0022-2836\(05\)80360-2](https://doi.org/10.1016/S0022-2836(05)80360-2)

[Get rights and content](#)

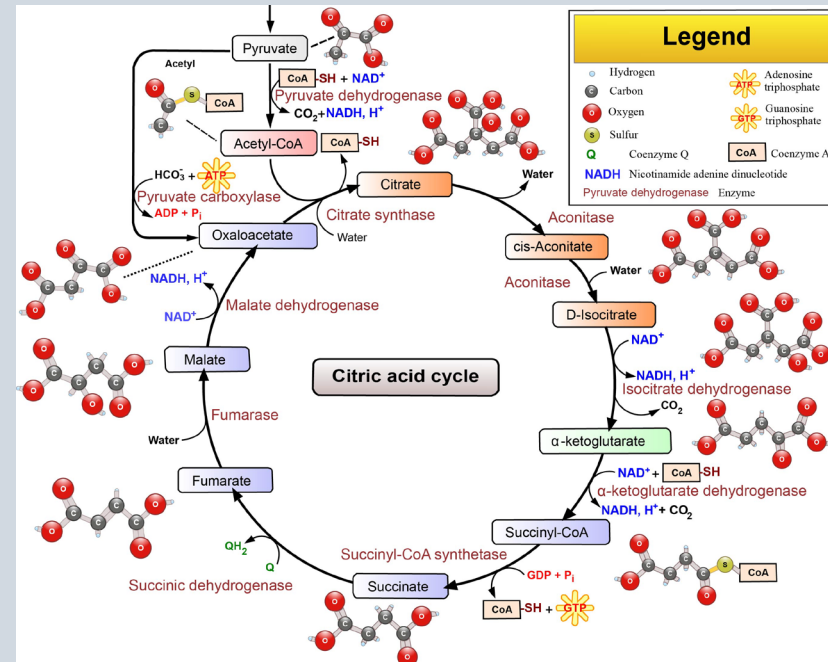
What would you say...you do here?

Create pipelines/workflows
(a set of data processing elements connected in series)



What would you say...you do here?

Create pipelines/workflows
(a set of data processing elements connected in series)



Bioinformatics mostly requires skills you have

Frequent literature searches for new methods

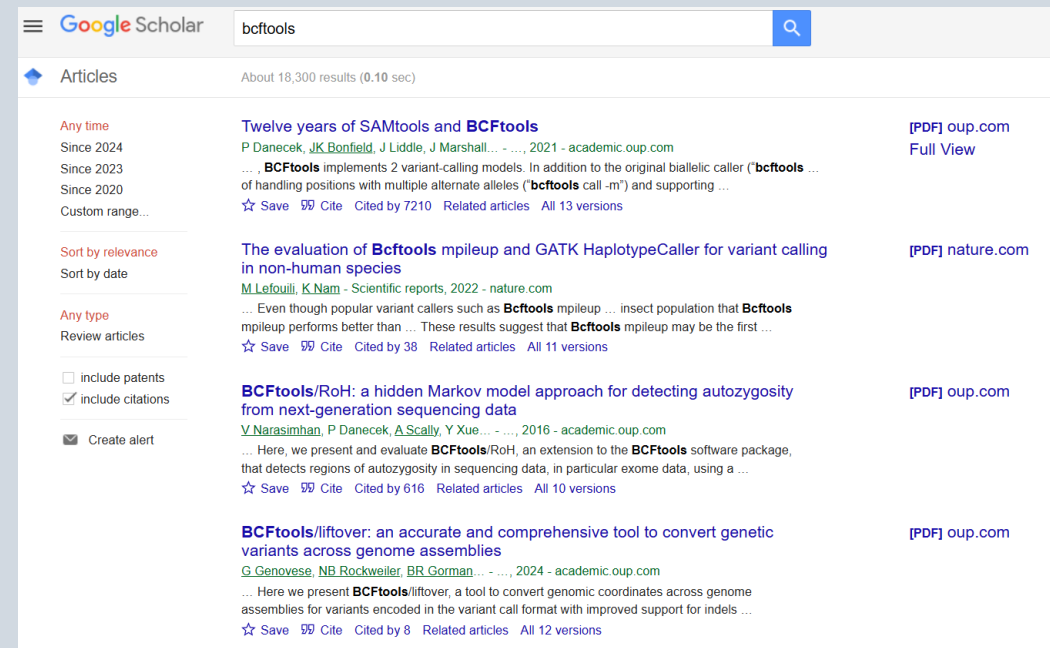
- Typically desire highly cited software
- Methods should be well maintained and documented

Reading documentation is required to understand the scope of the method

- Is it appropriate for your pipeline? Not always clear from the manuscript
- Do you have access to the required data? Not always possible, especially in non-model organisms

Downloading, installing, and executing software

- Straightforward if using conda/mamba, Docker, and Singularity



The screenshot shows a Google Scholar search for 'bcftools'. The search bar at the top contains 'bcftools' and a magnifying glass icon. Below the search bar, it says 'Articles' and 'About 18,300 results (0.10 sec)'. On the left side, there are filters for 'Any time' (with options: Since 2024, Since 2023, Since 2020, Custom range...), 'Sort by relevance' (with 'Sort by date' as an option), 'Any type' (with 'Review articles' as an option), and checkboxes for 'include patents' and 'include citations' (which is checked). There is also a 'Create alert' button. The main results area shows three articles:

- Twelve years of SAMtools and BCFtools**
P Danecek, JK Bonfield, J Liddle, J Marshall... - ..., 2021 - academic.oup.com
... , **BCFtools** implements 2 variant-calling models. In addition to the original biallelic caller ("bcftools ... of handling positions with multiple alternate alleles ("bcftools call -m") and supporting ...
☆ Save ⓘ Cite Cited by 7210 Related articles All 13 versions [PDF] oup.com Full View
- The evaluation of BCFtools mpileup and GATK HaplotypeCaller for variant calling in non-human species**
M Lefouili, K Nam - Scientific reports, 2022 - nature.com
... Even though popular variant callers such as **BCFtools** mpileup ... insect population that **BCFtools** mpileup performs better than ... These results suggest that **BCFtools** mpileup may be the first ...
☆ Save ⓘ Cite Cited by 38 Related articles All 11 versions [PDF] nature.com
- BCFtools/RoH: a hidden Markov model approach for detecting autozygosity from next-generation sequencing data**
V Narasimhan, P Danecek, A Scally, Y Xue... - ..., 2016 - academic.oup.com
... Here, we present and evaluate **BCFtools/RoH**, an extension to the **BCFtools** software package, that detects regions of autozygosity in sequencing data, in particular exome data, using a ...
☆ Save ⓘ Cite Cited by 616 Related articles All 10 versions [PDF] oup.com
- BCFtools/liftover: an accurate and comprehensive tool to convert genetic variants across genome assemblies**
G Genovese, NB Rockweiler, BR Gorman... - ..., 2024 - academic.oup.com
... Here we present **BCFtools/liftover**, a tool to convert genomic coordinates across genome assemblies for variants encoded in the variant call format with improved support for indels ...
☆ Save ⓘ Cite Cited by 8 Related articles All 12 versions [PDF] oup.com

Bioinformatics mostly requires skills you have

Frequent literature searches for new methods

- Typically desire highly cited software
- Methods should be well maintained and documented

Reading documentation is required to understand the scope of the method

- Is it appropriate for your pipeline? Not always clear from the manuscript
- Do you have access to the required data? Not always possible, especially in non-model organisms

Downloading, installing, and executing software

- Straightforward if using conda/mamba, Docker, and Singularity

bcftools(1) Manual Page

NAME

bcftools - utilities for variant calling and manipulating VCFs and BCFs.

SYNOPSIS

bcftools [-version|--version-only] [-help] [**COMMAND**] [**OPTIONS**]

DESCRIPTION

BCFtools is a set of utilities that manipulate variant calls in the Variant Call Format (VCF) and its binary counterpart BCF. All commands work transparently with both VCFs and BCFs, both uncompressed and BGZF-compressed.

Most commands accept VCF, bgzipped VCF and BCF with filetype detected automatically even when streaming from a pipe. Indexed VCF and BCF will work in all situations. Un-indexed VCF and BCF and streams will work in most, but not all situations. In general, whenever multiple VCFs are read simultaneously, they must be indexed and therefore also compressed. (Note that files with non-standard index names can be accessed as e.g. "bcftools view -r X:2928329 file.vcf.gz#idx##non-standard-index-name".)

BCFtools is designed to work on a stream. It regards an input file "-" as the standard input (stdin) and outputs to the standard output (stdout). Several commands can thus be combined with Unix pipes.

VERSION

This manual page was last updated **2024-04-29 08:11 BST** and refers to bcftools git version **1.20-6-g5977f1b3+**.

BCF1

The obsolete BCF1 format output by versions of samtools <= 0.1.19 is **not** compatible with this version of bcftools. To read BCF1 files one can use the view command from old versions of bcftools packaged with samtools versions <= 0.1.19 to convert to VCF, which can then be read by this version of bcftools.

samtools-0.1.19/bcftools/bcftools view file.bcf1 | bcftools view

VARIANT CALLING

See *bcftools call* for variant calling from the output of the *samtools mpileup* command. In versions of samtools <= 0.1.19 calling was done with *bcftools view*. Users are now required to choose between the old samtools calling model (*-c/--consensus-caller*) and the new multiallelic calling model (*-m/--multiallelic-caller*). The multiallelic calling model is recommended for most tasks.

FILTERING EXPRESSIONS

See [EXPRESSIONS](#)

LIST OF COMMANDS

For a full list of available commands, run **bcftools** without arguments. For a full list of available options, run **bcftools COMMAND** without arguments.

- **annotate** . edit VCF files, add or remove annotations
- **call** . SNP/indel calling (former "view")

Bioinformatics mostly requires skills you have

Frequent literature searches for new methods

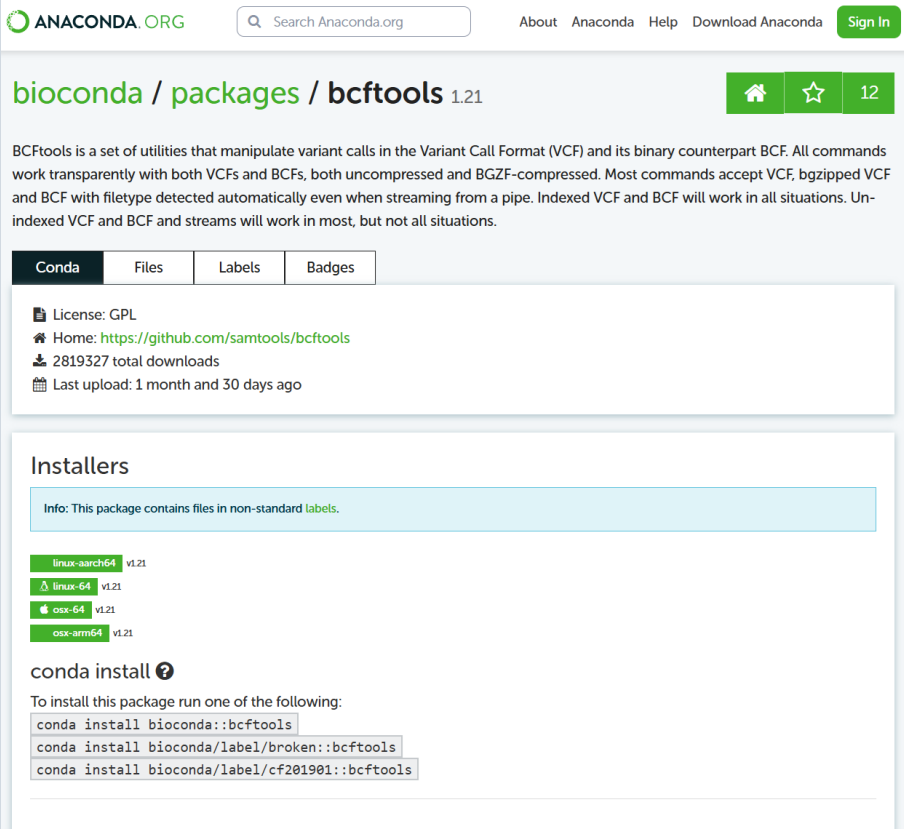
- Typically desire highly cited software
- Methods should be well maintained and documented

Reading documentation is required to understand the scope of the method

- Is it appropriate for your pipeline? Not always clear from the manuscript
- Do you have access to the required data? Not always possible, especially in non-model organisms

Downloading, installing, and executing software

- Straightforward if using conda/mamba, Docker, and Singularity



The screenshot shows the bioconda.org website. At the top is the ANACONDA.ORG logo and a search bar. Navigation links include About, Anaconda, Help, Download Anaconda, and Sign In. The main heading is 'bioconda / packages / bcftools 1.21'. Below this is a description of BCFTools as a set of utilities for manipulating variant calls. A tabbed interface shows 'Conda' as the active tab, with options for Files, Labels, and Badges. Under the Conda tab, details include the GPL license, a GitHub link, 281,932 total downloads, and the last upload date. An 'Installers' section contains a warning about non-standard labels and a list of platform-specific installers (linux-aarch64, linux-64, osx-64, osx-arm64). At the bottom, a 'conda install' command is provided with a help icon and a note to run one of the following commands, followed by three example commands for different environments.

ANACONDA.ORG Search Anaconda.org About Anaconda Help Download Anaconda Sign In

bioconda / packages / bcftools 1.21

BCFTools is a set of utilities that manipulate variant calls in the Variant Call Format (VCF) and its binary counterpart BCF. All commands work transparently with both VCFs and BCFs, both uncompressed and BGZF-compressed. Most commands accept VCF, bgzipped VCF and BCF with filetype detected automatically even when streaming from a pipe. Indexed VCF and BCF will work in all situations. Un-indexed VCF and BCF and streams will work in most, but not all situations.

Conda Files Labels Badges

License: GPL
Home: <https://github.com/samtools/bcftools>
2819327 total downloads
Last upload: 1 month and 30 days ago

Installers

Info: This package contains files in non-standard labels.

linux-aarch64 v1.21
linux-64 v1.21
osx-64 v1.21
osx-arm64 v1.21

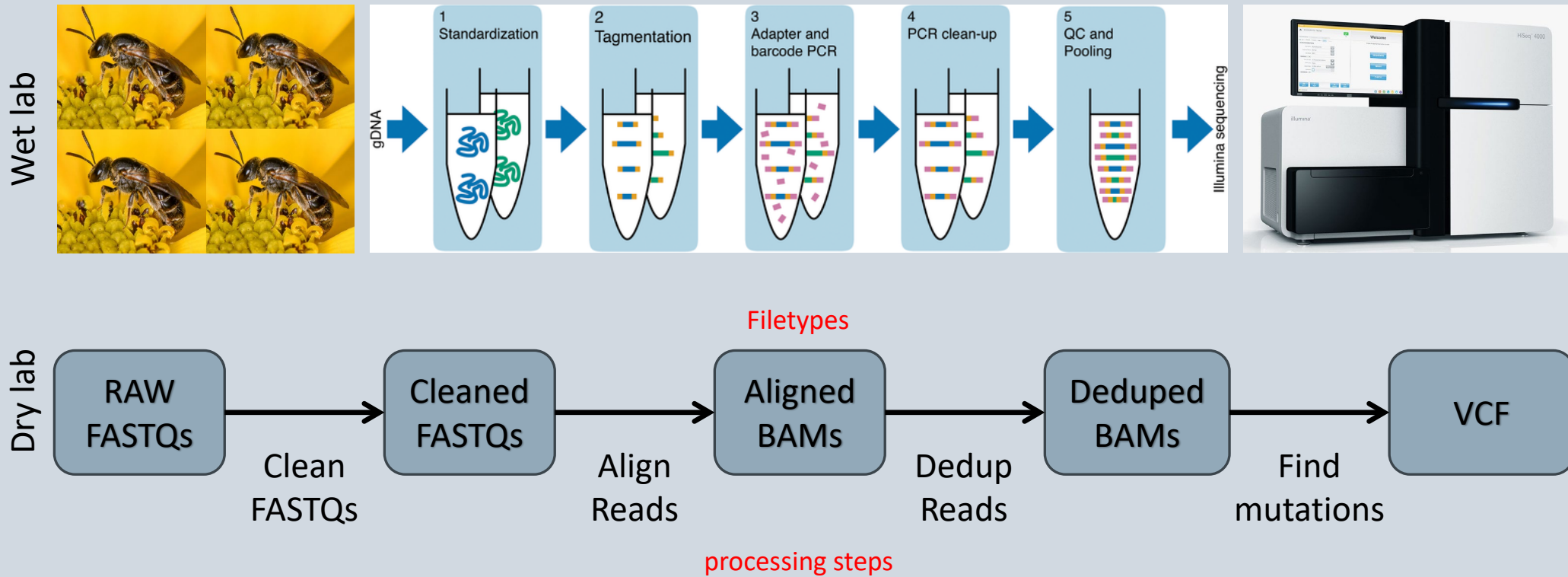
conda install ?

To install this package run one of the following:

```
conda install bioconda::bcftools
conda install bioconda/label/broken::bcftools
conda install bioconda/label/cf201901::bcftools
```

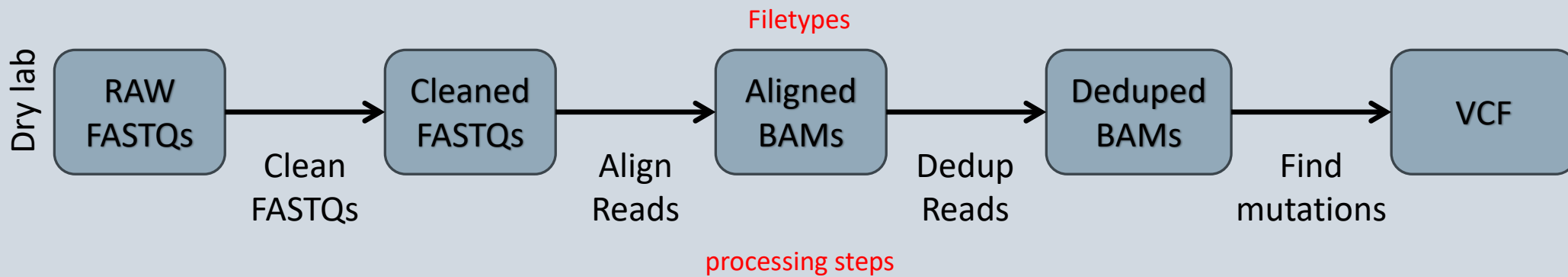
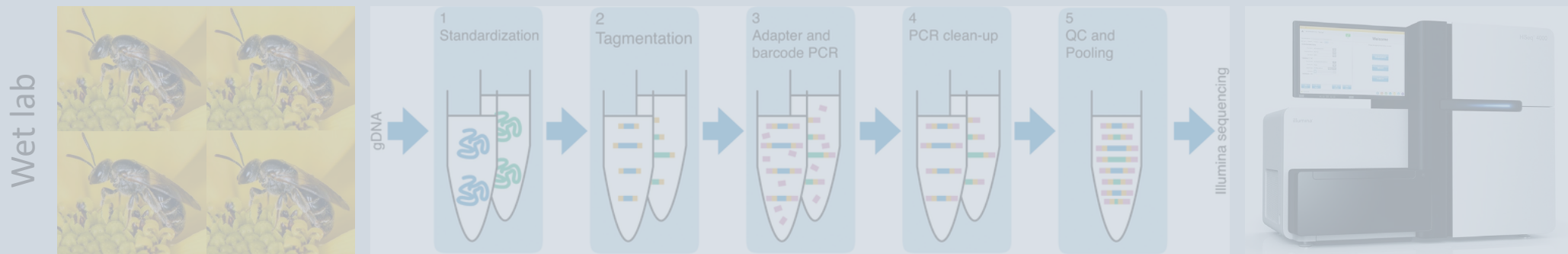
Resequencing pipeline: overview

Sequencing multiple samples to discover mutations

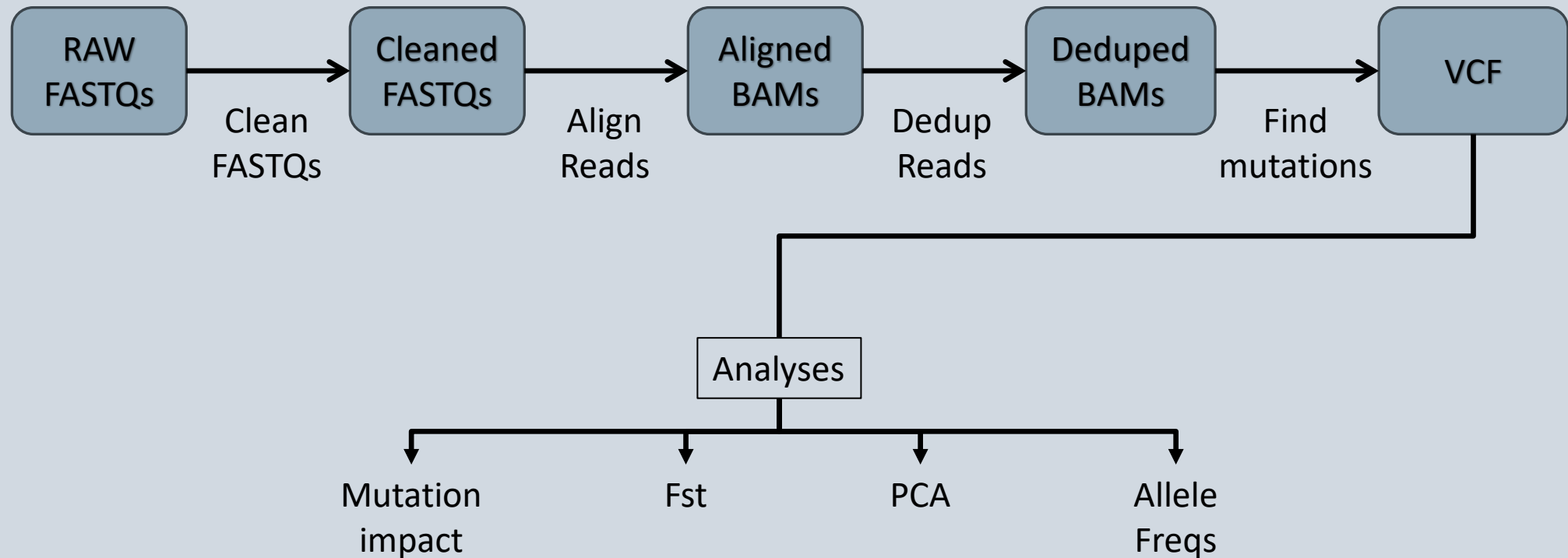


Resequencing pipeline: overview

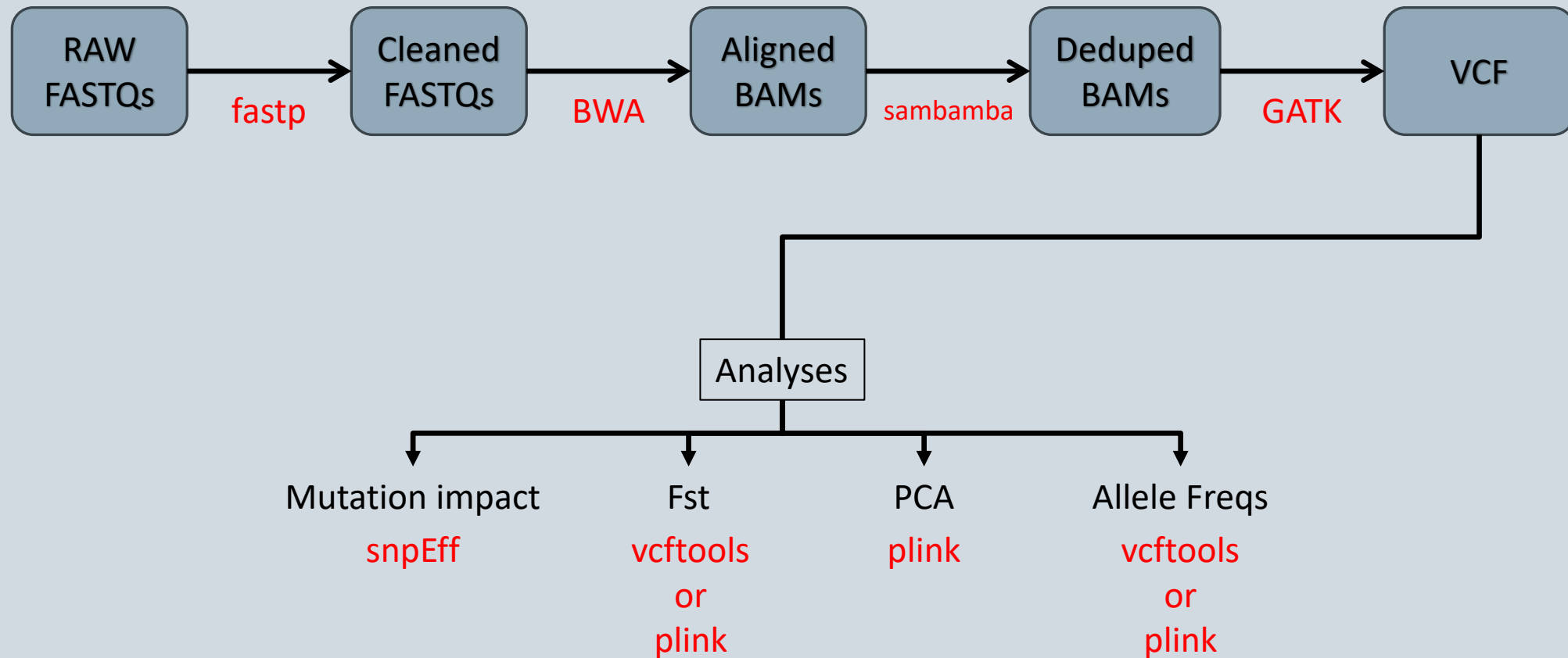
Sequencing multiple samples to discover mutations



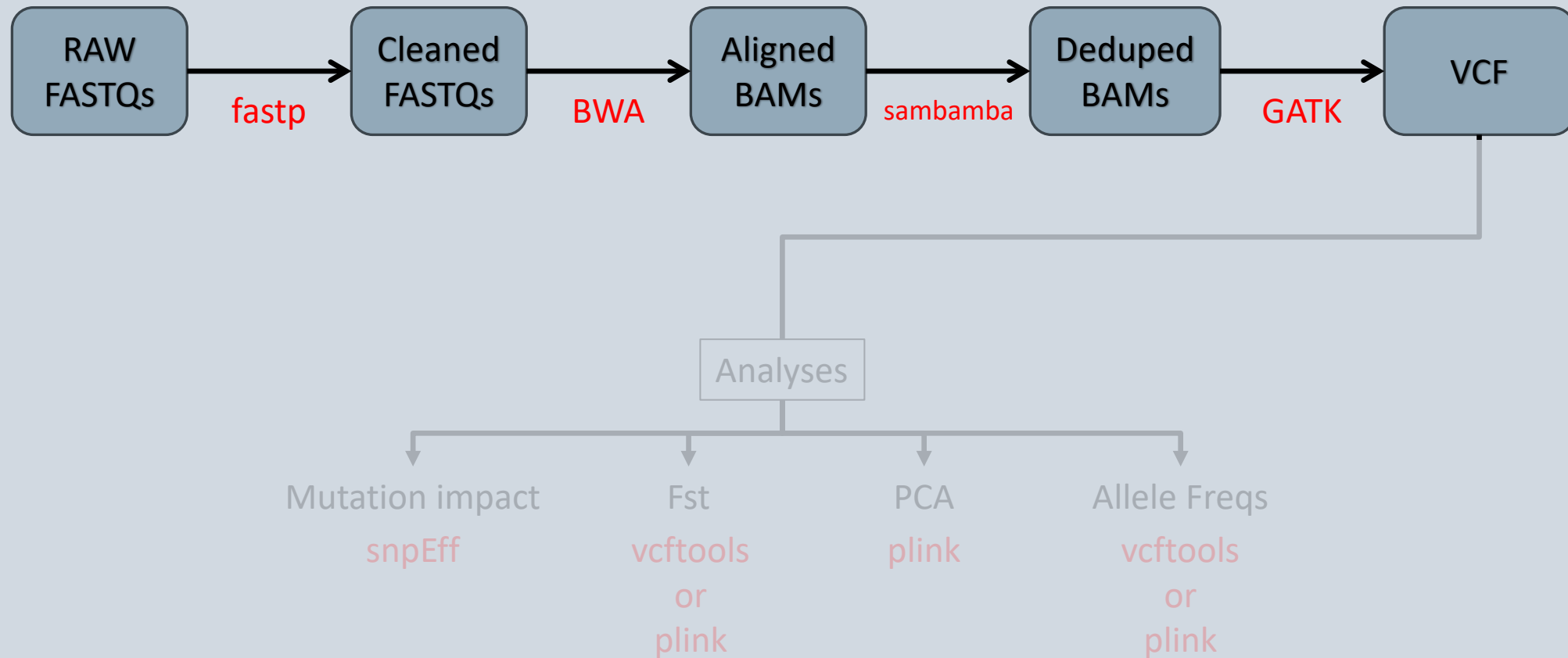
Resequencing pipeline: mutation analysis



Resequencing pipeline: the **programs**



Resequencing pipeline: the **programs**



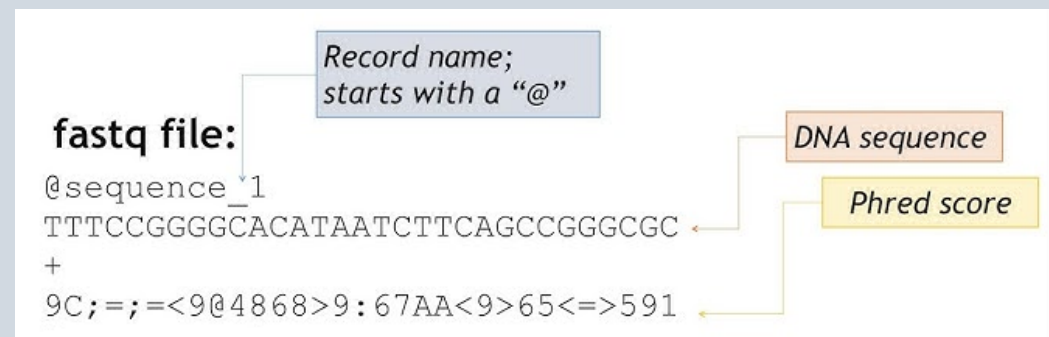
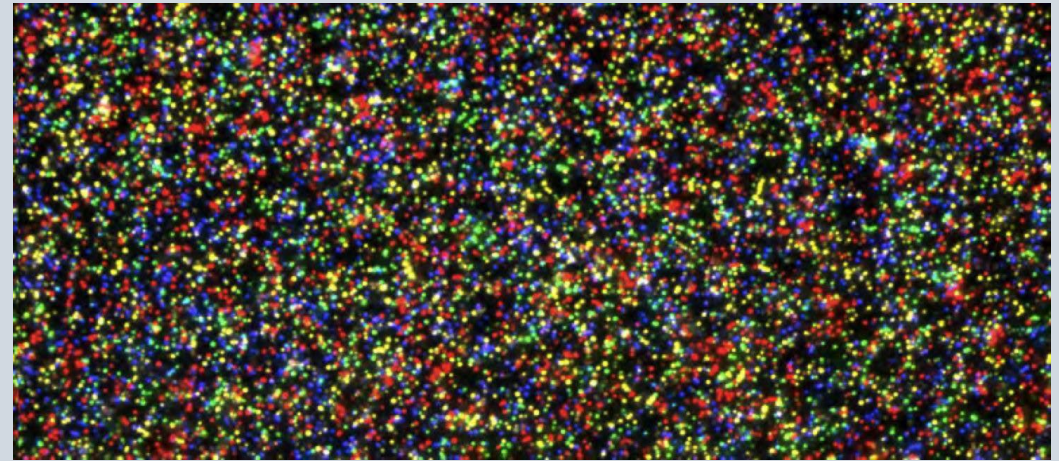
FASTQ file

Converts the colors viewed by the sequencer

Each record is from a single read or “dot”

A FASTQ record includes 4 parts

- The header: @sequence_1
- The sequence
- A separator
- The Phred quality score for each base



FASTQ file

Phred uses ASCII characters to represent the probability of an incorrect base call

- 9: 0.004
- C: 0.0004

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

fastq file:

@sequence_1

TTTCCGGGGCACATAATCTTCAGCCGGGCGC

+

9C;;=<9@4868>9:67AA<9>65<=>591

Record name;
starts with a "@"

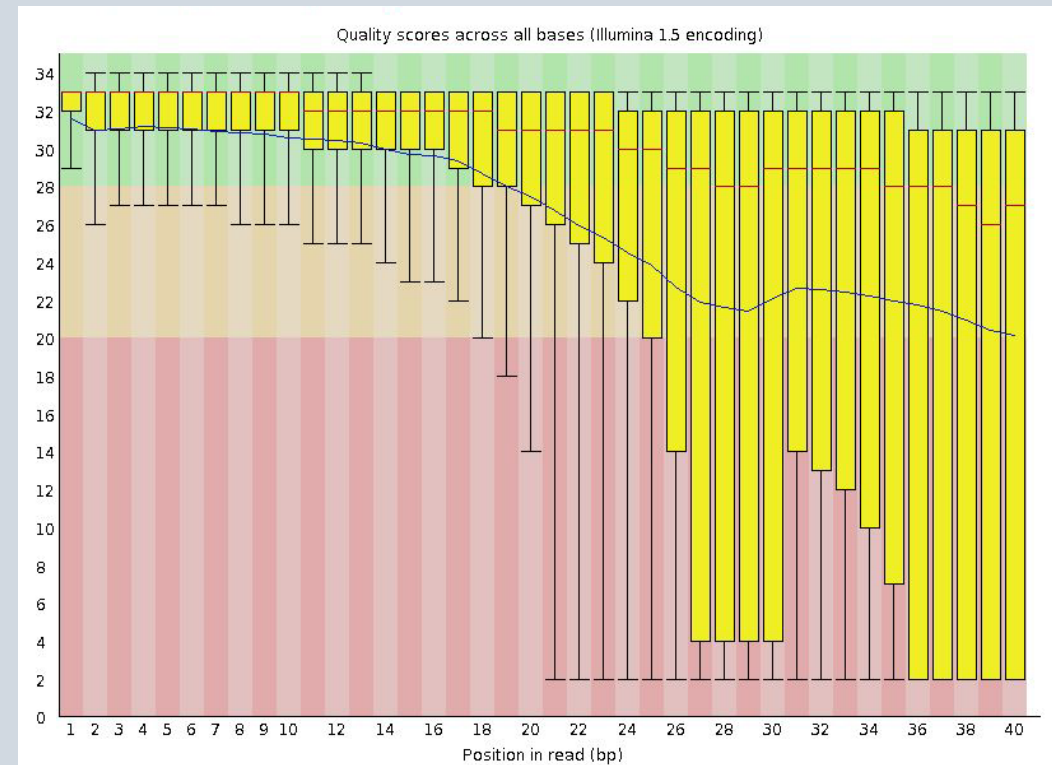
DNA sequence

Phred score

Cleaned FASTQ file

Use Phred quality scores to clean reads

First create a report using FASTQC



FASTQC Report
(Raw FASTQ)

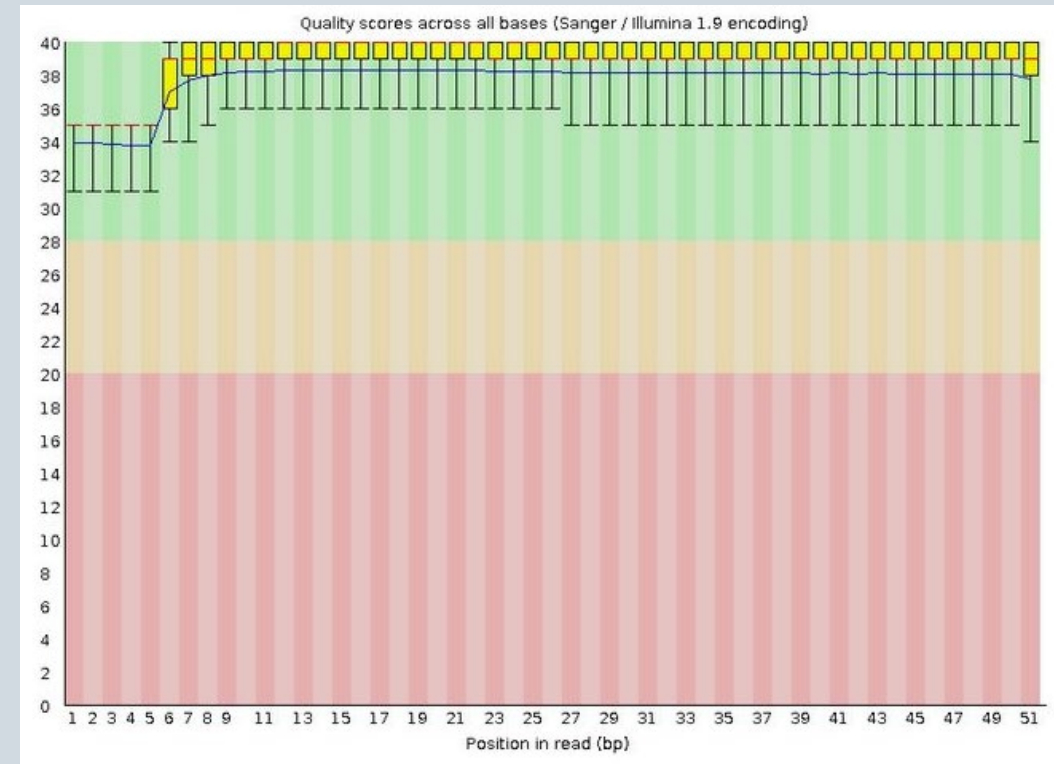
Cleaned FASTQ file

Trim nucleotides from reads

- Low quality nucleotides, often at the end of reads
- Adapter sequences
- Poly-A tails (if RNAseq data)

Popular programs for cleaning FASTQs

- fastp



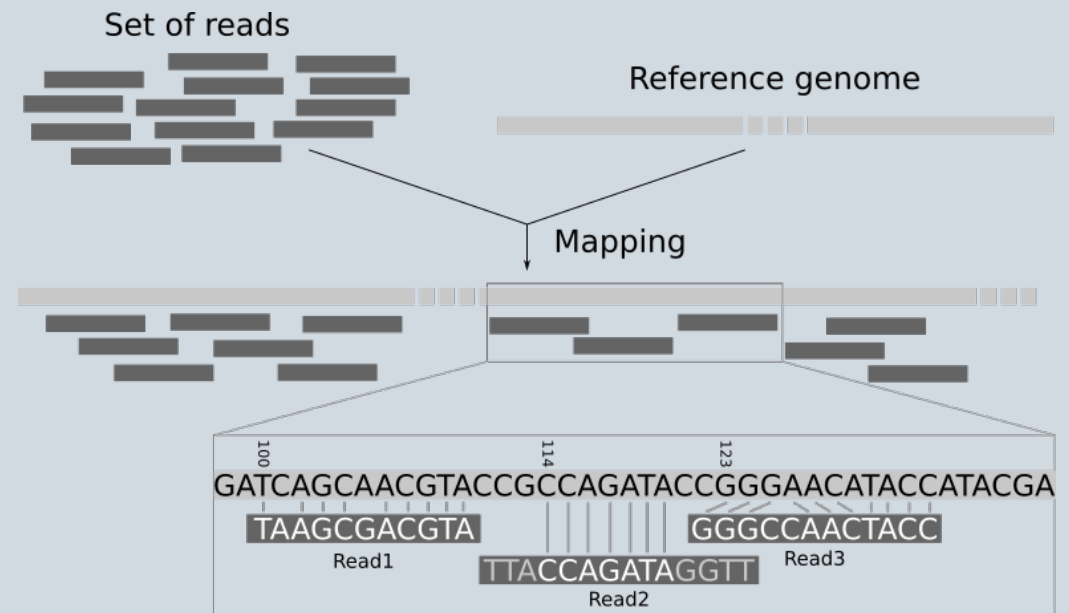
FASTQC Report
(Cleaned FASTQ)

BAM file

Stores reads aligned to a reference genome

Ideally the reads and reference should be from the same species

- May use different species but alignment becomes more difficult



Reads aligned based on **similarity** to positions in reference genome

BAM file

A BAM file begins with a header

- Begin with @

Records includes details on

- Where the read aligned
- Details on type of read (paired, mapped, unmapped)
- Sequence and quality

Popular programs

- BWA (DNA)
- STAR (RNA)

```
@HD VN:1.6 SO:coordinate
@SQ SN:ref LN:45
r001 99 ref 7 30 8M2I4M1D3M = 37 39 TTAGATAAAGGATACTG *
r002 0 ref 9 30 3S6M1P1I4M * 0 0 AAAAGATAAGGATA *
r003 0 ref 9 30 5S6M * 0 0 GCCTAAGCTAA * SA:Z:ref,29,-,6H5M,17,0;
r004 0 ref 16 30 6M14N5M * 0 0 ATAGCTTCAGC *
r003 2064 ref 29 17 6H5M * 0 0 TAGGC * SA:Z:ref,9,+,5S6M,30,1;
r001 147 ref 37 30 9M = 7 -39 CAGCGGCAT * NM:i:1
```

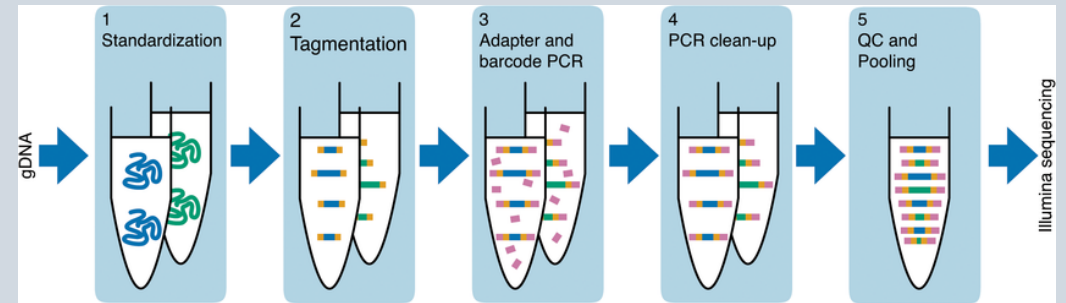
read id chrom pos

sequence

quality

Deduped BAM file

BAM files are expected to consist of reads randomly sampled from across the genome

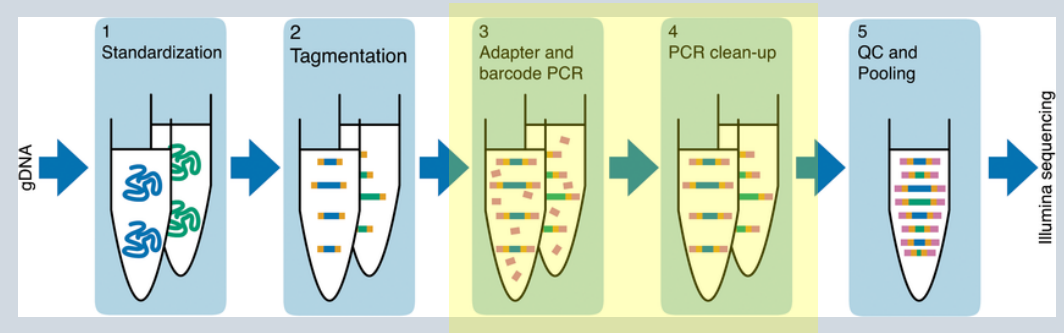


Deduped BAM file

BAM files are expected to consist of reads randomly sampled from across the genome

However, in our protocol we used PCR

- Increase amount of DNA
- Enrich DNA fragments with adapters (and barcodes)
- Results in PCR duplicates

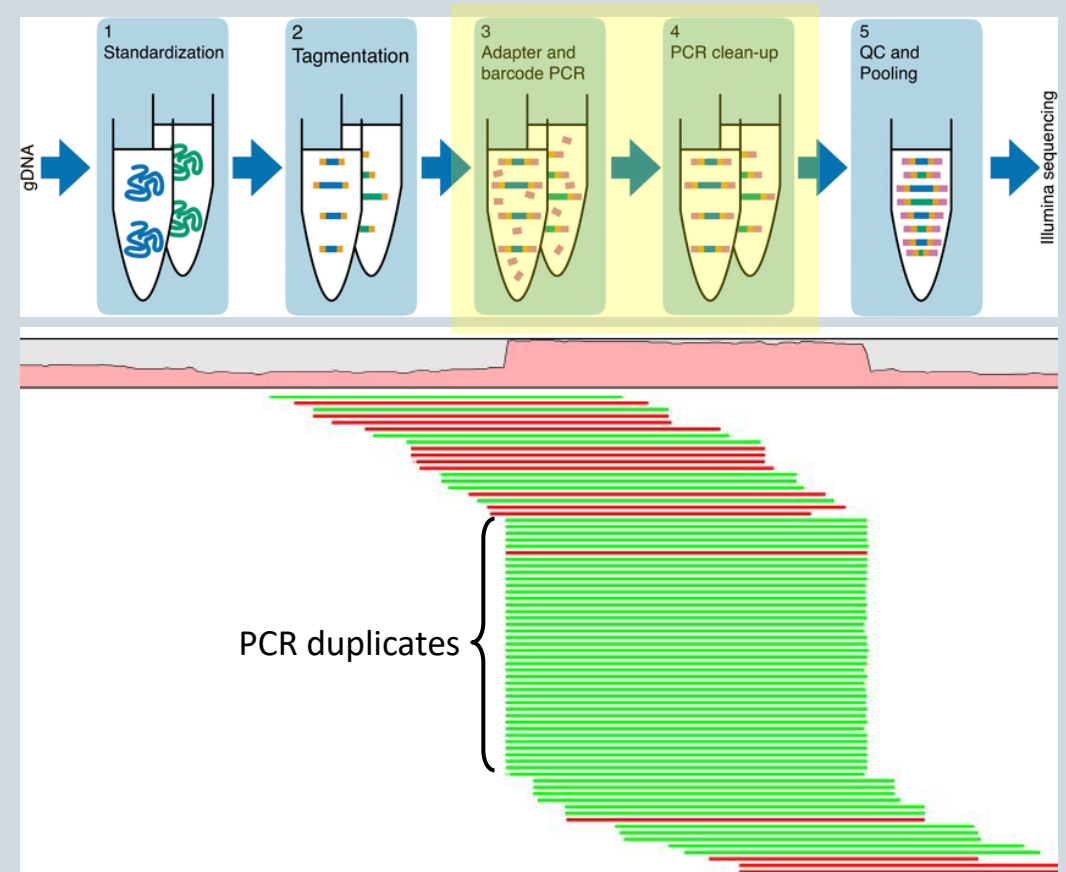


Deduped BAM file

PCR duplicates must be identified to allow subsequent programs to account for them

Popular programs to dedup

- SAMBAMBA
- PICARD
- SAMTOOLS



VCF file (Variant Call Format)

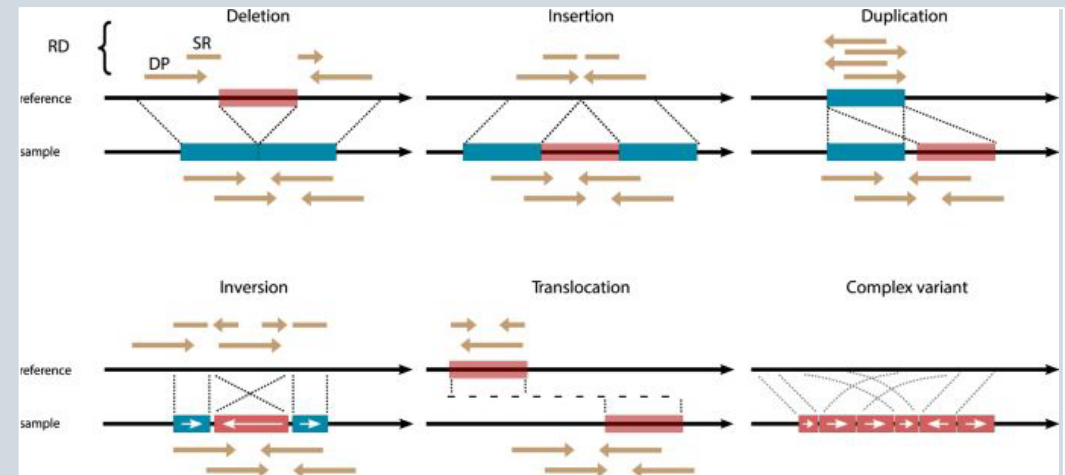
Uses reference aligned reads to detect variants

- Single nucleotide polymorphisms (SNPs)
- Indels

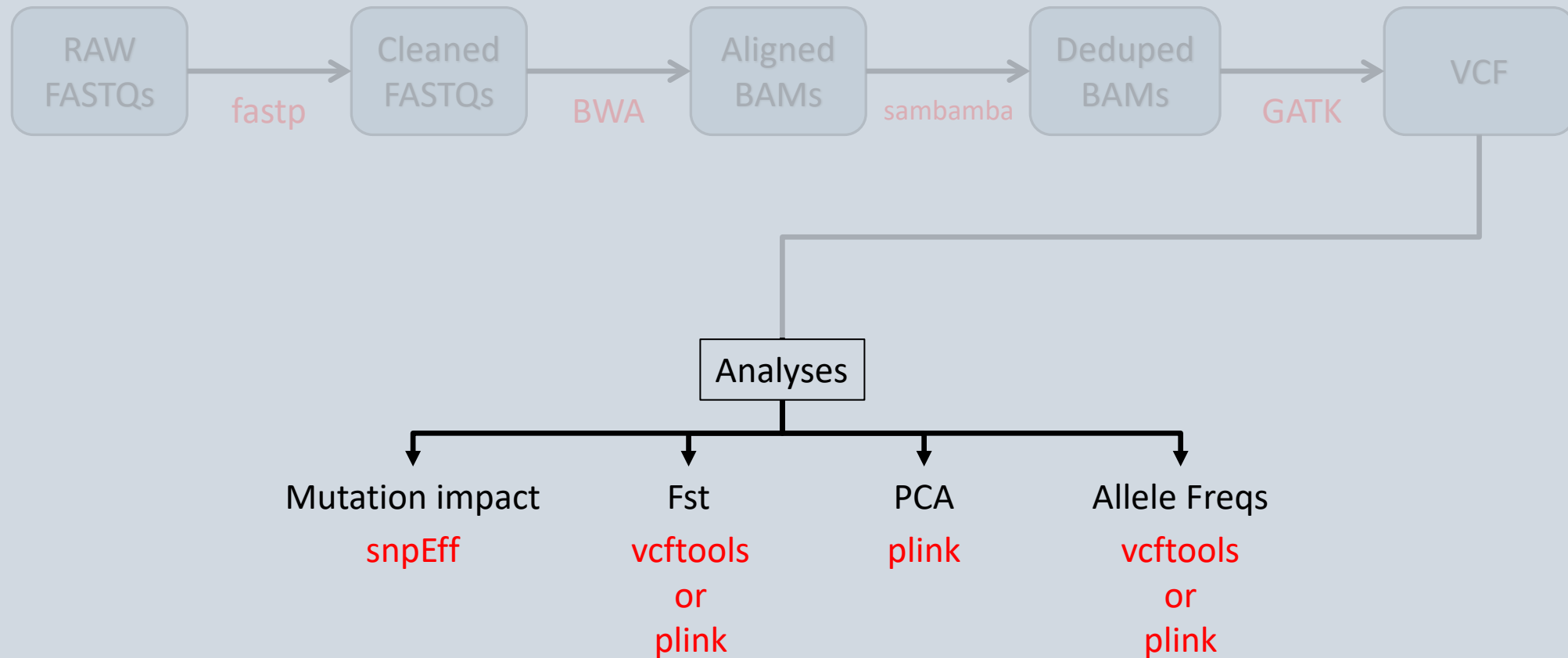
Popular programs to create VCFs

- SNPs
 - GATK (gold standard)
 - Freebayes
- Indels, etc.
 - DELLY
 - LUMPY
 - MANTA

```
##fileformat=VCFv4.1
##FILTER=<ID=PASS,Description="All filters passed">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Phased Genotype">
##contig=<ID=11>
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT 49 50 51 52
11 5498142 * G A . PASS . GT 1|1 1|1 1|1 1|1
11 5498159 * TG T . PASS . GT 0|0 0|0 0|0 0|0
11 5498334 * G A . PASS . GT 1|1 0|0 0|0 0|1
11 5498551 * G C . PASS . GT 0|0 0|1 0|1 1|0
11 5498649 * G C . PASS . GT 0|0 0|0 0|1 0|0
11 5498683 * A G . PASS . GT 1|1 1|1 1|1 1|1
```



Resequencing pipeline: the **programs**



VCF file: brief description

```
##fileformat=VCFv4.1
##FILTER=<ID=PASS,Description="All filters passed">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Phased Genotype">
##contig=<ID=11>
#CHROM  POS  ID   REF  ALT  QUAL  FILTER  INFO  FORMAT  49  50  51  52
11  5498142  *    G   A   .    PASS   .    GT      1|1 1|1 1|1 1|1
11  5498159  *    TG  T   .    PASS   .    GT      0|0 0|0 0|0 0|0
11  5498334  *    G   A   .    PASS   .    GT      1|1 0|0 0|0 0|1
11  5498551  *    G   C   .    PASS   .    GT      0|0 0|1 0|1 1|0
11  5498649  *    G   C   .    PASS   .    GT      0|0 0|0 0|1 0|0
11  5498683  *    A   G   .    PASS   .    GT      1|1 1|1 1|1 1|1
```

Critical format to understand. Many analyses use VCF files

VCF file: brief description

Header

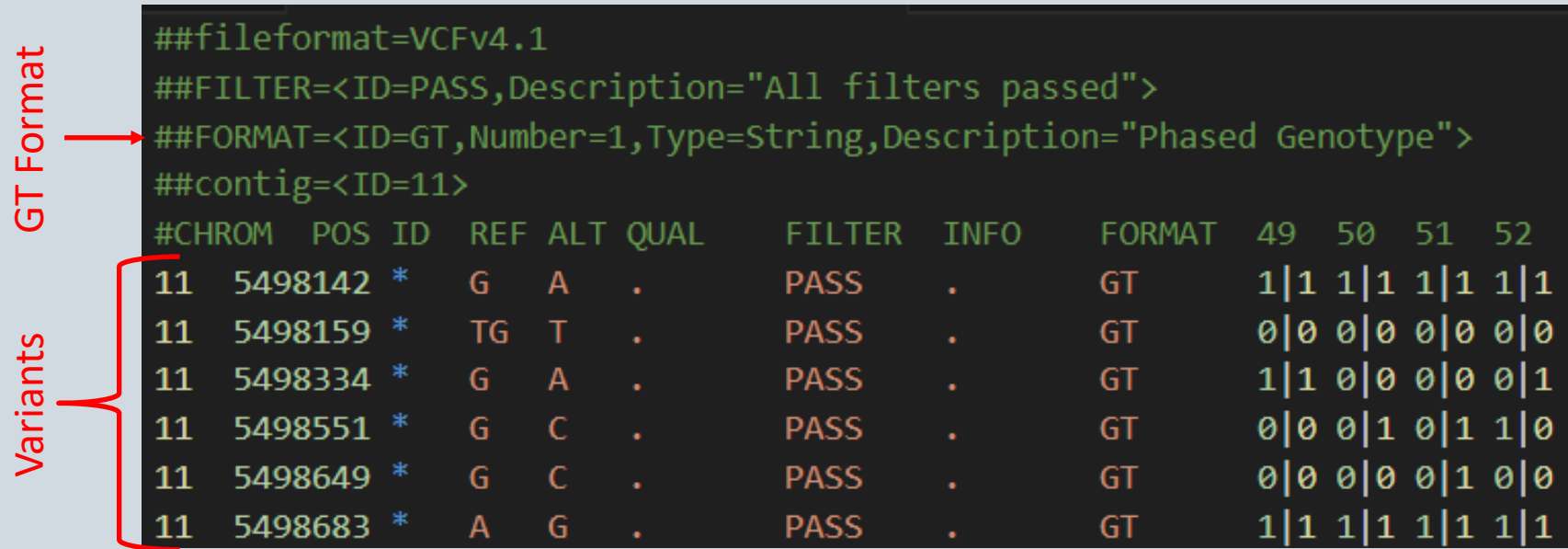
```
##fileformat=VCFv4.1
##FILTER=<ID=PASS,Description="All filters passed">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Phased Genotype">
##contig=<ID=11>
#CHROM  POS  ID   REF  ALT  QUAL  FILTER  INFO  FORMAT  49  50  51  52
11  5498142  *    G   A   .    PASS   .    GT     1|1 1|1 1|1 1|1
11  5498159  *    TG  T   .    PASS   .    GT     0|0 0|0 0|0 0|0
11  5498334  *    G   A   .    PASS   .    GT     1|1 0|0 0|0 0|1
11  5498551  *    G   C   .    PASS   .    GT     0|0 0|1 0|1 1|0
11  5498649  *    G   C   .    PASS   .    GT     0|0 0|0 0|1 0|0
11  5498683  *    A   G   .    PASS   .    GT     1|1 1|1 1|1 1|1
```

##: Contain definitions of abbreviations used throughout file (most not shown)

#: The primary header, gives the column names for each variant

- The chromosome, position, ID, reference allele, the alternative allele(s), and quality score
- Filter and info tags
- The format column details how to read the sample columns
- The samples

VCF file: brief description



```
##fileformat=VCFv4.1
##FILTER=<ID=PASS,Description="All filters passed">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Phased Genotype">
##contig=<ID=11>
#CHROM  POS  ID   REF  ALT  QUAL  FILTER  INFO  FORMAT  49  50  51  52
11  5498142  *   G   A   .   PASS   .   GT   1|1 1|1 1|1 1|1
11  5498159  *   TG  T   .   PASS   .   GT   0|0 0|0 0|0 0|0
11  5498334  *   G   A   .   PASS   .   GT   1|1 0|0 0|0 0|1
11  5498551  *   G   C   .   PASS   .   GT   0|0 0|1 0|1 1|0
11  5498649  *   G   C   .   PASS   .   GT   0|0 0|0 0|1 0|0
11  5498683  *   A   G   .   PASS   .   GT   1|1 1|1 1|1 1|1
```

How to read the GT format

- 0|0 phased vs 0/0 unphased
- 0/0 are homozygotes for the REF allele
- 1/1 are homozygotes for the ALT allele
- 0/1 are heterozygotes – i.e. have both alleles

VCF file: brief description

```
##fileformat=VCFv4.1
##FILTER=<ID=PASS,Description="All filters passed">
##FILTER=<ID=LowQual,Description="Low quality">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Phased Genotype">
##contig=<ID=11>
#CHROM  POS  ID  REF ALT  QUAL  FILTER  INFO  FORMAT  49  50  51  52  53
11  100000  *  C   A   .   q10    .   GT      0|1 0|0 0|0 0|1 0|0
11  100001  *  A   C   .   PASS   .   GT      0|1 0|1 0|0 0|0 0|0
11  100002  *  T   C   .   q10    .   GT      0|1 0|1 0|0 0|0 0|0
11  100003  *  T   C   .   PASS   .   GT      0|1 0|1 0|0 0|0 0|0
11  100004  *  G   C   .   LowQual .   GT      0|1 0|1 0|0 0|0 0|0
11  100005  *  C   T   .   PASS   .   GT      0|1 0|1 0|0 0|0 0|0
11  100006  *  T   G   .   s50    .   GT      ./ . 0|1 ./ . ./ . 1|1
```

Can use programs like VCFtools for quality control

- Drop variants with quality issues (LowQual)
- Drop variants with too many missing samples (s50)

VCF file analysis: intro

We have access to a simple VCF file of tusked and tuskless elephants: elephants_long.vcf

Q1: How many variants do we have within our elephant dataset

- Count the number of variants, get the line count and exclude the header

```
grep -v '#' elephants_long.vcf | wc -l
```

A1: 27597

Q2: What are the names of the samples in our VCF?

- Get the last line of the header, which starts with: #CHROM

```
grep '#CHROM' elephants_long.vcf
```

A2:

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT
0045B	2981B	2982B	2983B	2984B	2985B	2986A	G13	G15
G16	G17A	G18A	G19A	G20A	G21AG22A	T2B		

VCF file analysis: Fst

Let's use vcftools to calculate Fst in sliding windows along the first chromosome

- FST is calculated between two groups of individuals
- If the groups are from different species, Fst will be high
- If the groups are from the same species, Fst will often be lower

To calculate Fst we need to tell vcftools which samples in our VCF file correspond to these groups

tusked.txt

```
2981B
2982B
2985B
G18A
```

tuskless.txt

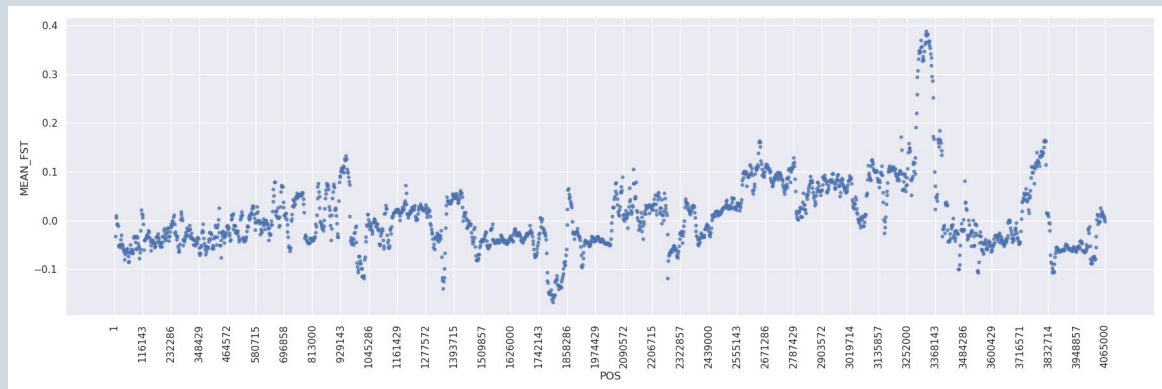
```
0045B
2983B
2984B
2986A
G17A
G19A
G22A
```

VCF file analysis: Fst

Let's calculate Fst between the two groups using the following command:

```
vcftools \  
--vcf elephants_long.vcf \  
--weir-fst-pop tusked.txt \  
--weir-fst-pop tuskless.txt \  
--fst-window-size 10000 \  
--fst-window-step 2000 \  
--out fst
```

CHROM	BIN_START	BIN_END	N_VARIANTS	WEIGHTED_FST	MEAN_FST
scaffold_0	1	10000	26	-0.0459742	-0.0323767
scaffold_0	2001	12000	25	0.00754489	0.00676232
scaffold_0	4001	14000	22	0.0155885	0.0105999
scaffold_0	6001	16000	28	-0.0109385	-0.00638111
scaffold_0	8001	18000	31	-0.0115322	-0.00669053
scaffold_0	10001	20000	28	-0.0202694	-0.0102851
scaffold_0	12001	22000	26	-0.0537565	-0.0517176
scaffold_0	14001	24000	30	-0.053488	-0.050845
scaffold_0	16001	26000	27	-0.0497459	-0.04946



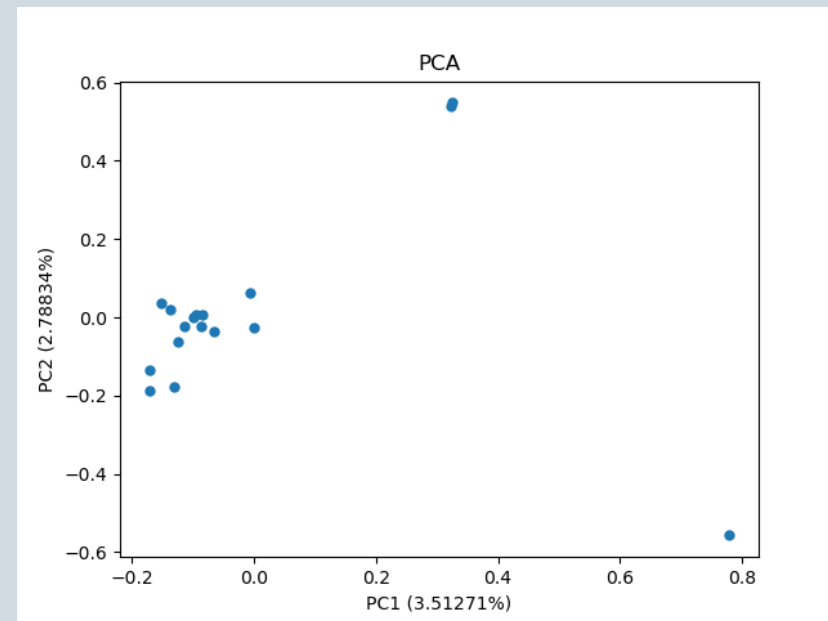
Plot of MEAN_FST

VCF file analysis: PCA

Let's calculate PCA using the following command:

```
plink --vcf elephants_long.vcf --pca 2 --allow-extra-chr --out pca
```

0045B	0045B	0.321888	0.54021
2981B	2981B	-0.000314435	-0.025959
2982B	2982B	-0.137171	0.0209842
2983B	2983B	-0.0647252	-0.0345524
2984B	2984B	-0.171759	-0.187896
2985B	2985B	-0.169974	-0.133993
2986A	2986A	0.324437	0.548219
G13	G13	-0.129983	-0.176786
G15	G15	-0.0873685	-0.0219399
G16	G16	-0.151139	0.0365722
G17A	G17A	-0.0998264	-0.000493901
G18A	G18A	-0.114367	-0.021457
G19A	G19A	0.779483	-0.557152
G20A	G20A	-0.124635	-0.0628551
G21A	G21A	-0.0844193	0.00763434
G22A	G22A	-0.095012	0.00597826
T2B	T2B	-0.00574599	0.0637649



Plot of last two columns

VCF file analysis: allele frequencies

Let's calculate allele frequencies using the following command:

```
plink --vcf elephants_long.vcf --freq counts --allow-extra-chr --out freq
```

frequencies

CHR	SNP	A1	A2	MAF	NCHROBS
scaffold_0	.	T	A	0.2647	34
scaffold_0	.	G	A	0.2647	34
scaffold_0	.	T	C	0.08824	34
scaffold_0	.	G	T	0.08824	34
scaffold_0	.	T	C	0.2647	34
scaffold_0	.	C	G	0.2647	34
scaffold_0	.	A	T	0.08824	34

counts

CHR	SNP	A1	A2	C1	C2	G0
scaffold_0	.	T	A	9	25	0
scaffold_0	.	G	A	9	25	0
scaffold_0	.	T	C	3	31	0
scaffold_0	.	G	T	3	31	0
scaffold_0	.	T	C	9	25	0
scaffold_0	.	C	G	9	25	0
scaffold_0	.	A	T	3	31	0

There were 17 diploid samples in this VCF, or 34 chromosomes

- If we examine the counts, we can identify the minor allele
- For the first variant, the minor allele is T with a count of 9
- The MAF can then be calculated as $9/34 = 0.2647$

VCF file analysis: annotations

```
#CHROM POS ID REF ALT QUAL FILTER
7 117227832 . G T . .

INFO
AC 14
AN 22
ANN T|stop_gained|HIGH|CFTR|ENSG00000001626|transcript|ENST00000003084|protein_coding|12/27|c.1624G>T|p.Gly542*|1756/6128|1624/4443|542/1480||
ANN T|stop_gained|HIGH|CFTR|ENSG00000001626|transcript|ENST00000454343|protein_coding|11/26|c.1441G>T|p.Gly481*|1573/5949|1441/4260|481/1419||
LOF (CFTR|ENSG00000001626|11|0.27)
NMD (CFTR|ENSG00000001626|11|0.27)
```

VCFs store annotations within the INFO column

- Once added other programs can access the information
- *Please note the text is stored as a single line (separated by commas)*

Popular programs

- SnpEff

VCF file analysis: exporting

Let's create a simple table using the following command:

```
gatk VariantsToTable -V elephants_long.vcf -F CHROM -F POS -F TYPE -GF GT -O elephants_long.table
```

CHROM	POS	TYPE	0045B.GT	2981B.GT	2982B.GT	2983B.GT
scaffold_0	145	SNP	A/A	A/A	A/T	A/T
scaffold_0	396	SNP	A/A	A/A	A/G	A/G
scaffold_0	412	SNP	C/C	C/C	C/C	C/T
scaffold_0	530	SNP	C/C	C/C	C/T	C/T
scaffold_0	538	SNP	G/G	G/G	G/C	G/C
scaffold_0	784	INDEL	G/G	G/G	G/GC	G/GC
scaffold_0	1153	SNP	A/A	A/A	A/G	A/G
scaffold_0	1202	SNP	G/G	G/G	G/A	G/A

Other kinds of sequencing data

DNA: methods

Genome assembly

- Often done using a combination of technologies
 - Long reads (10kbp+) allows sequencing through repeats, but less accurate
 - HiC provides genomic rearrangement information
 - Short reads (150bp) cannot sequence repetitive content, but more accurate

Restriction-associated DNA (RADseq)

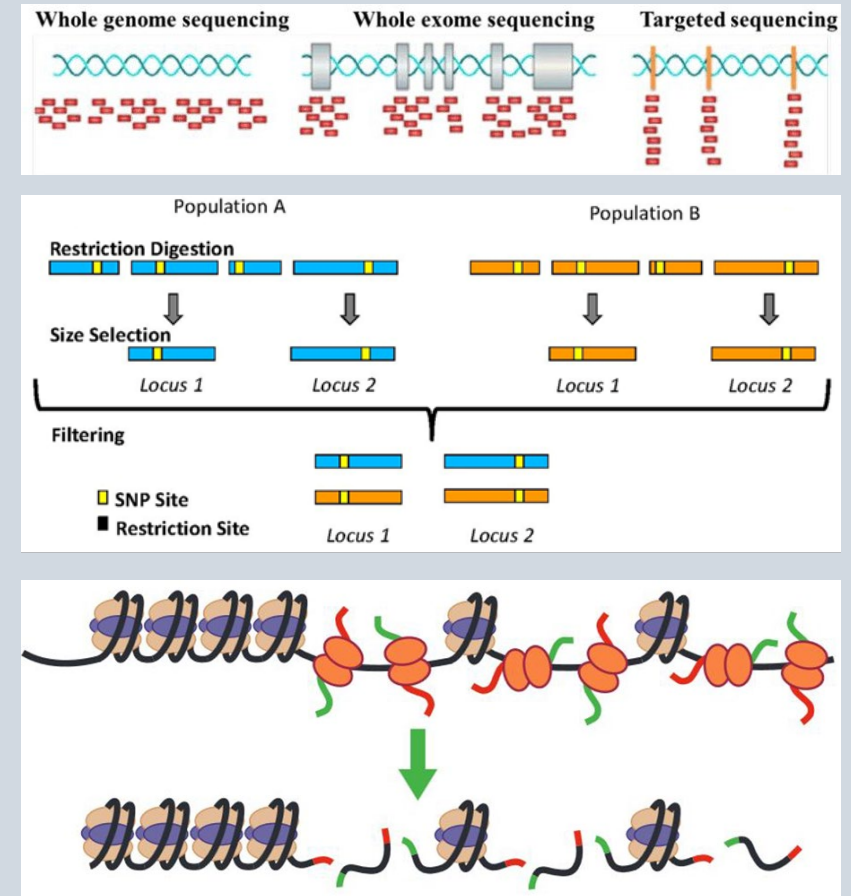
- Sequence DNA near restriction enzyme site
- Popular method in population genetics

ATAC-seq

- Sequence open chromatin regions of the genome

CHIP-seq

- Sequence regions bound to transcription factors and other proteins



RNA: methods

RNAseq

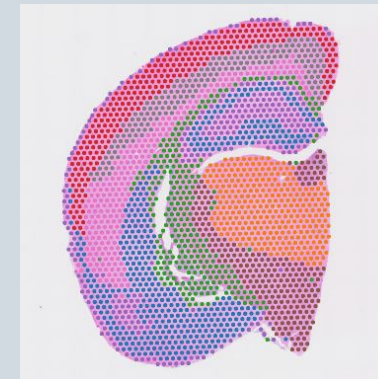
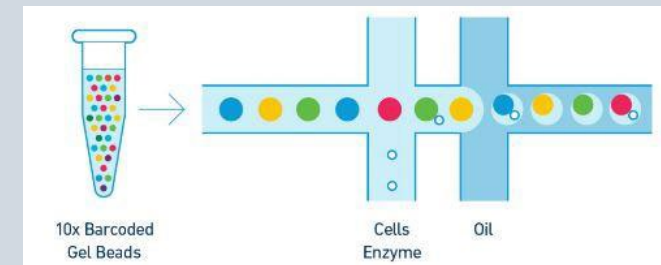
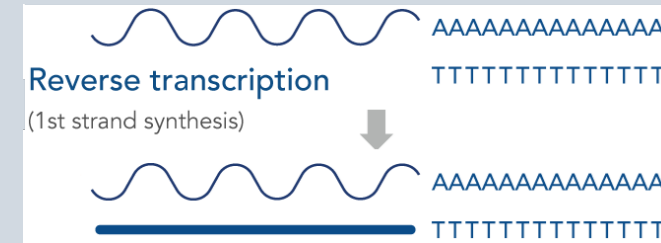
- Used for differential expression analyses and to construct gene regulatory networks
- Reads are reported as DNA due to the conversion of RNA to cDNA

Single-cell RNAseq

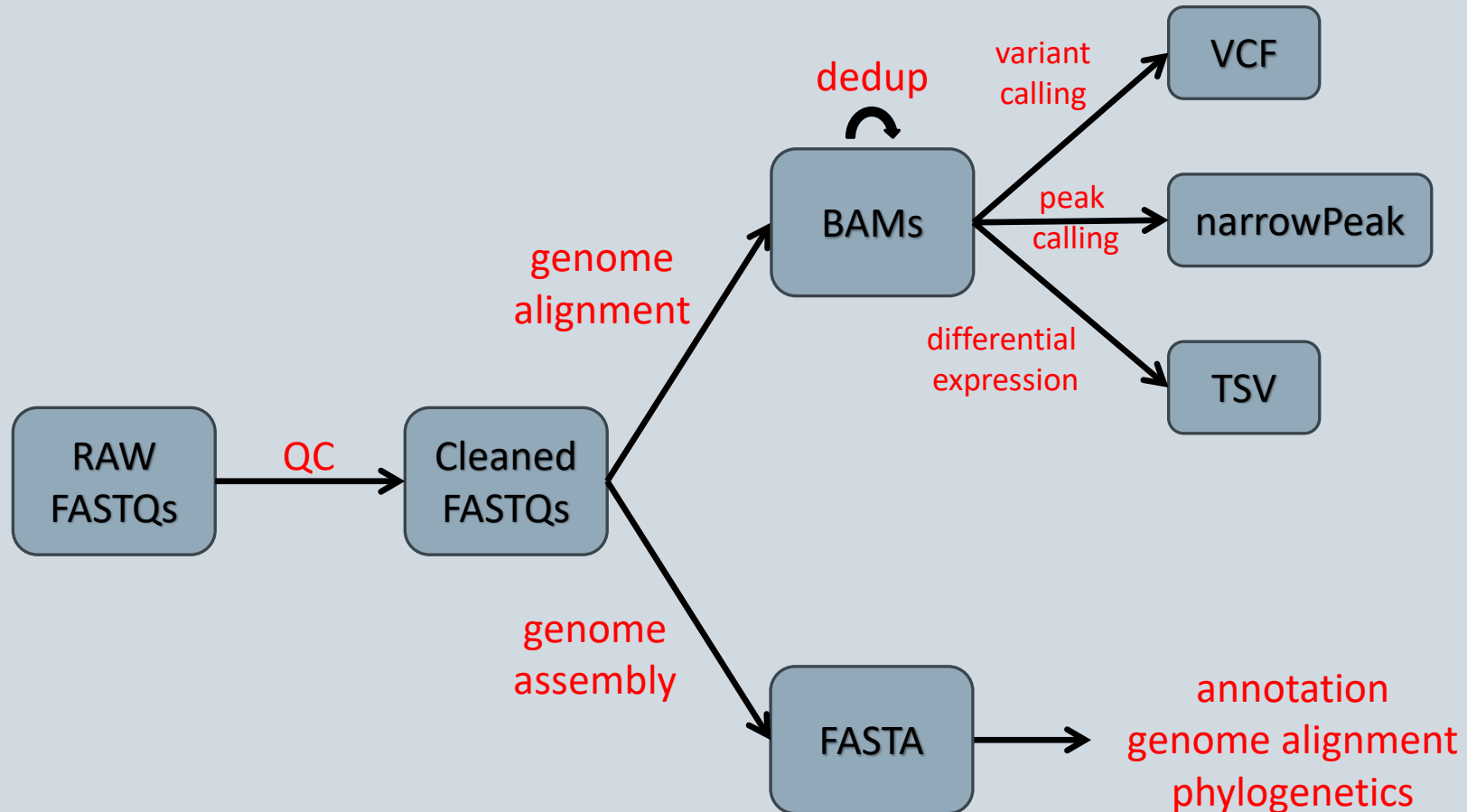
- Used to characterize the expression of specific cell types

Spatial transcriptomics

- Identify tissues and inter-tissue communication



Simple bioinformatic roadmap



RNAseq pipeline

