



# Module 5 - Séance 3

## Mapping & Variant Calling

Thierry Grange - Université Paris-Diderot  
Olivier Rué - INRA

# Objectifs

Processus d'analyse de données de séquences, des filtres de qualité à la détection de variants :

- SNVs et indels de petite taille

# Cluster de l'IFB

L'Institut Français de Bioinformatique met à disposition de la communauté un cluster de calculs

**Your turn! Se connecter au cluster**

## **# Sous Windows avec MobaXterm**

Session : ssh

Host : core.cluster.france-bioinformatique.fr

Specify username : coché et complété

## **# Sous Mac avec Cyberduck**

Open connexion : SFTP

Server : core.cluster.france-bioinformatique.fr

Username/Password : à compléter

# Cluster de l'IFB

```
# Chargement de l'environnement dédié à cette session  
$ module load conda  
$ source activate eba2018_variant_calling_python3
```

# Jeux de données : SNVs/Indels

Depuis que l'homme fait de l'élevage, il essaie de faire en sorte de toujours améliorer sa **production, que ce soit en quantité ou en qualité.**

Les technologies de génotypage permettent maintenant de **sélectionner les mâles reproducteurs en fonction du fond génétique** qu'ils vont pouvoir transmettre à leur descendance.

Chez le bovin, il existe un locus de caractères quantitatifs (QTL) lié à la production de lait, situé sur le **chromosome 6**, et plus exactement sur une région de 700 kb, composée de 7 gènes.



# Jeux de données : SNVs/Indels

Les échantillons **QTL+** sont caractérisés par une **diminution de la production en lait** et une augmentation des concentrations en protéine et lipide.

Vous aurez à votre disposition :

- Un extrait des données de séquences d'un échantillon du projet 1000 génomes bovins, phénotypé comme **QTL-** : **SRR1262731**
- Les résultats du variant calling pour deux échantillons phénotypés **QTL+** : **SRR1205992** et **SRR1205973**

**Your turn !**

**Quelle mutation est responsable de ce QTL ?**

# Emplacement des données brutes

- Jeux de données : SNVs/Indels

→ /shared/data/projects/du\_bii\_2019/data/module5/seance3

# Le Mapping



# Mapping short reads to a reference genome

- Définition: Prédiction du locus d'où vient une lecture
- Résultats: Liste de(s) (la) region(s) la(les) plus probable(s) avec une probabilité associée.
- Difficultés:

1. Gérer efficacement des centaines de millions de lecture, en utilisant l'information de la probabilité que la séquence soit correcte.

@SEQ\_ID

GATTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT

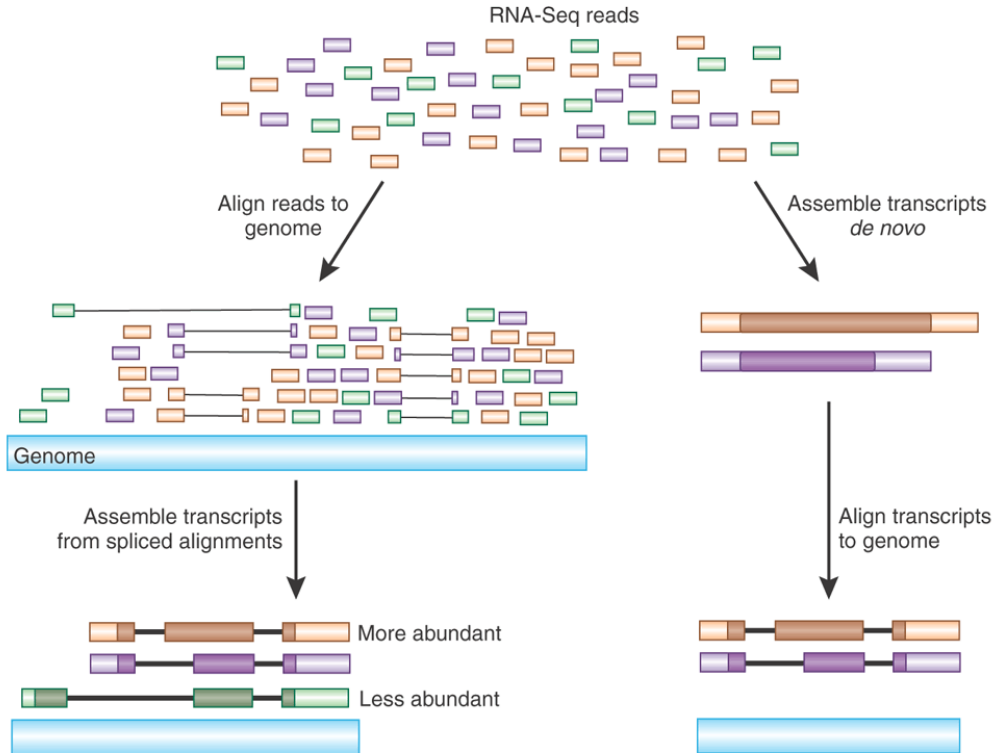
+

!"\*(((\*\*\*+))%%%++)(%%%%).1\*\*\*-+\*)"\*\*55CCF>>>>>CCCCCCC65

$$Q = -10 \log_{10} p$$

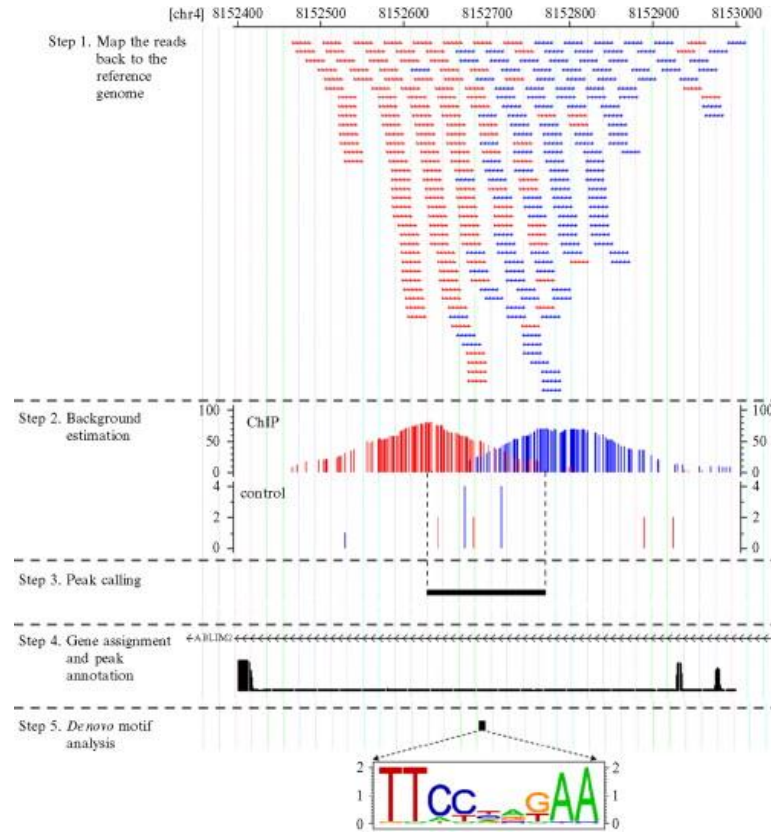
# Différentes situations initiales ont différentes solutions optimales

## RNA-seq



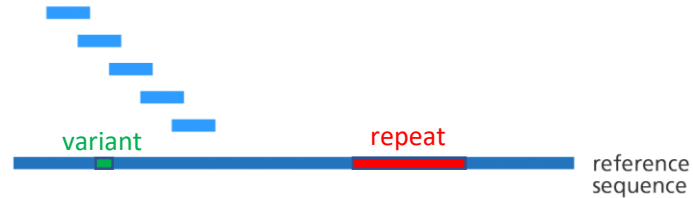
# Différentes situations initiales ont différentes solutions optimales

## ChIP-seq



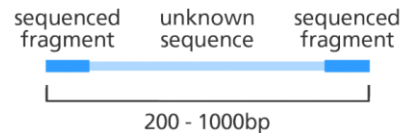
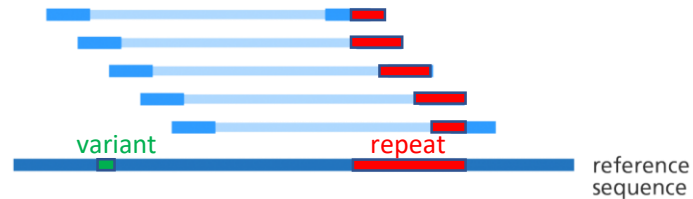
# Différentes situations initiales ont différentes solutions optimales

Single-end reads



## Resequencing

Paired-end reads



# Local or global alignments

GAAGCTCTAGGATTACGATCTTGATCGCCGGGAAATTATGATCCTGACCTGAGTTTAAGGCATGGACCCATAA

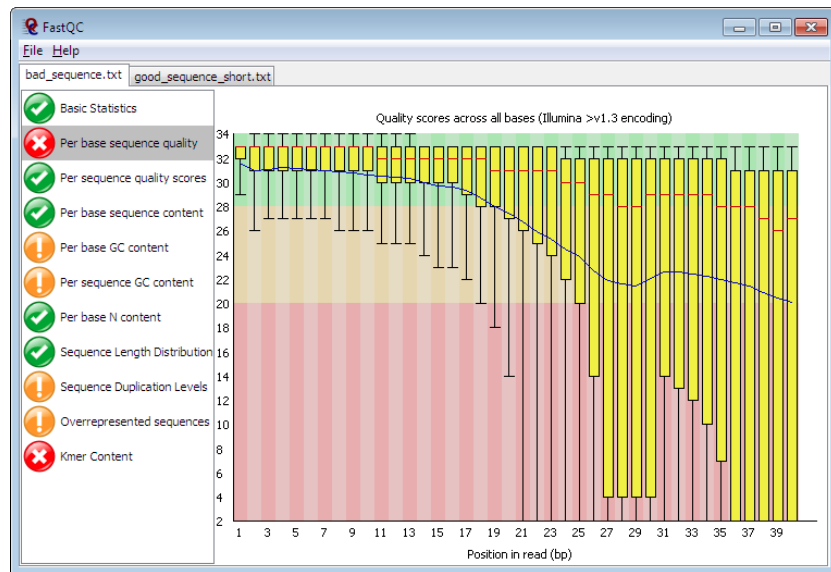
ATCTTGATCGCCGAC----ATT

GLOBAL

ATCTTGATCGCCGAC**CATT**

LOCAL, with soft clipping

For proper global alignment, it is important to do a proper previous read trimming



# Local or global alignments

GAAGCTCTAGGATTACGATCTTGATCGCCGGGAAATTATGATCCTGACCTGAGTTTAAGGCATGGACCCATAA

ATCTTGATCGCCGAC----ATT

GLOBAL

ATCTTGATCGCCGAC***ATT***

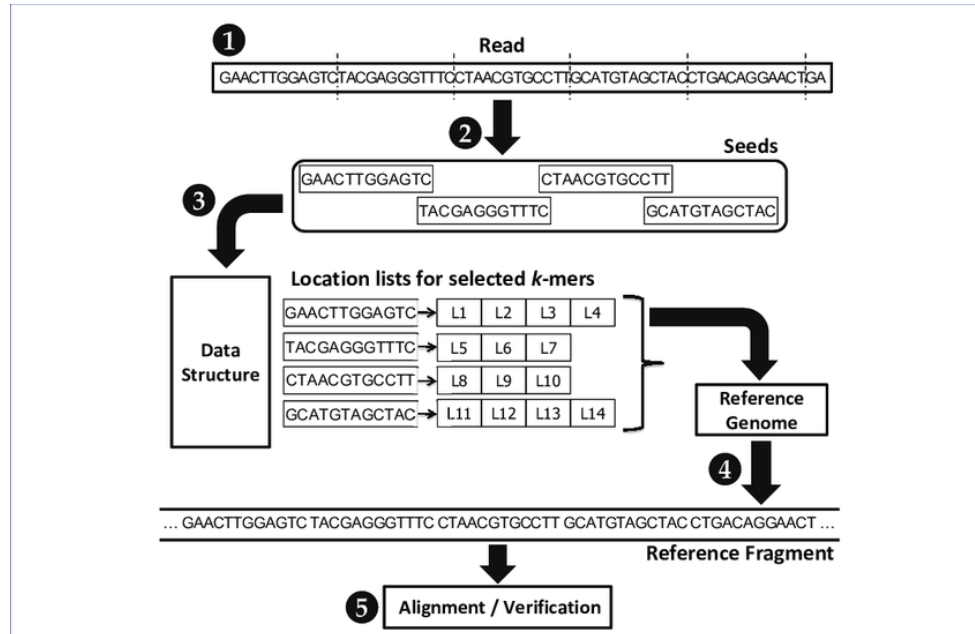
LOCAL, with soft clipping

Soft clipped  
sequences can be  
displayed as such  
on a browser



# Mapping: algorithm to be as fast and as precise as possible

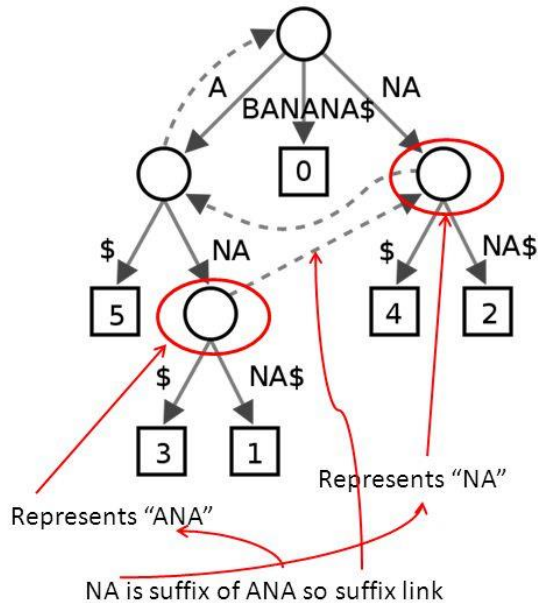
## Seed and extend



Plus la graine est petite, plus on a de chance de trouver la cible correcte mais plus le mapping sera lent

# Mapping: algorithm to be as fast and as precise as possible

## Traditional Sequence Alignment – Suffix Tree



Suffix tree for the string BANANA.

Each substring is terminated with special character \$.

The six paths from the root to a leaf (shown as boxes) correspond to the six suffixes

A\$,  
NA\$,  
ANA\$,  
NANA\$,  
ANANA\$ and  
BANANA\$.

The numbers in the leaves give the start position of the corresponding suffix.

Suffix links drawn dashed.

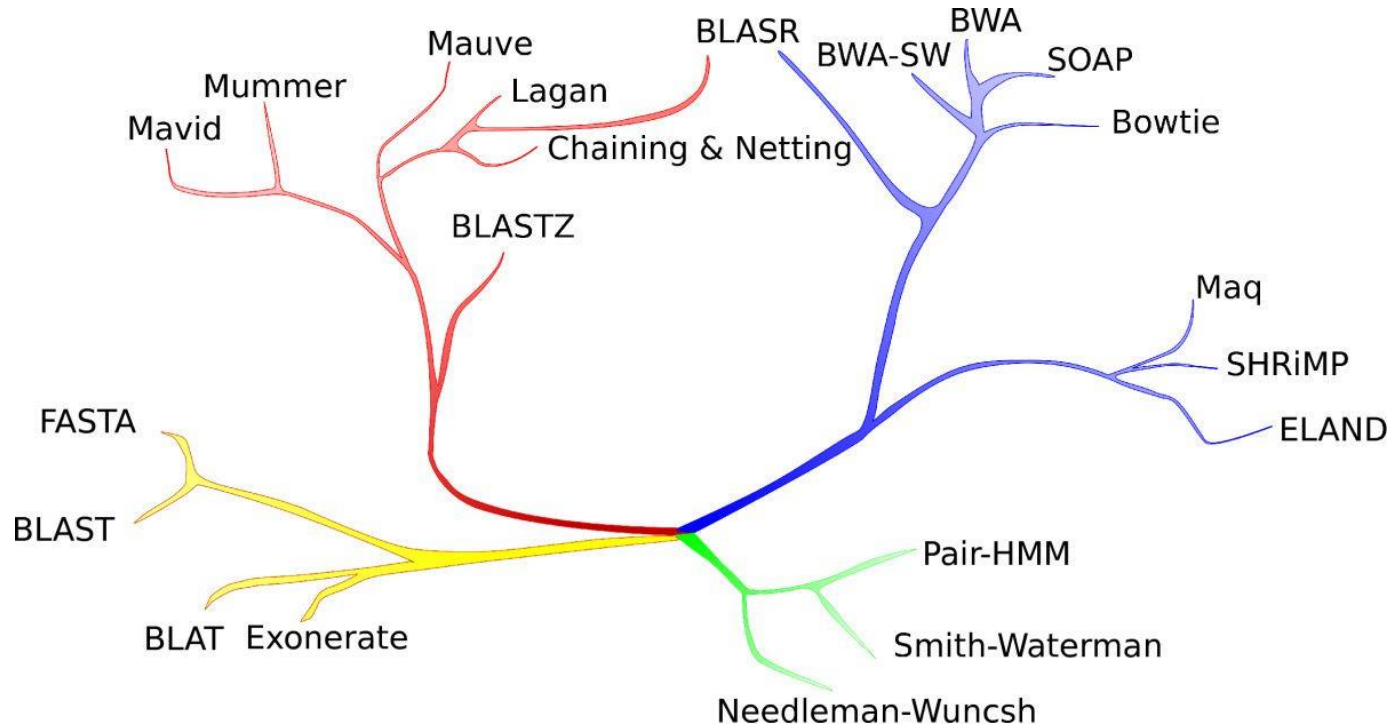


Mapping: algorithm to be as fast and as precise as possible

Compression: transformation de Burrow Wheeler

Transformation				
Input	All Rotations	Sorting All Rows in Alphabetical Order by their first letters	Taking Last Column	Output Last Column
<div><div>^BANANA  </div></div>	<div><div>^BANANA  </div><div>  ^BANANA</div><div>A   ^BANAN</div><div>NA   ^BANA</div><div>ANA   ^BAN</div><div>NANA   ^BA</div><div>ANANA   ^B</div><div>BANANA   ^</div></div>	<div><div>ANANA   ^B</div><div>ANA   ^BAN</div><div>A   ^BANAN</div><div>BANANA   ^</div><div>NANA   ^BA</div><div>NA   ^BANA</div><div>^BANANA  </div><div>  ^BANANA</div></div>	<div><div>ANANA   ^B</div><div>ANA   ^BAN</div><div>A   ^BANAN</div><div>BANANA   ^</div><div>NANA   ^BA</div><div>NA   ^BANA</div><div>^BANANA  </div><div>  ^BANANA</div></div>	<div><div>BNN^AA   A</div></div>

# Les familles d'algorithmes de mapping et d'alignement



# Bwa: un mapper puissant, précis et versatile

Usage: `bwa <command> [options]`

Command: index	index sequences in the FASTA format	
mem	BWA-MEM algorithm	local
fastmap	identify super-maximal exact matches	
pemerge	merge overlapping paired ends (EXPERIMENTAL)	
aln	gapped/ungapped alignment	global
samse	generate alignment (single ended)	
sampe	generate alignment (paired ended)	
bwasw	BWA-SW for long queries	
shm	manage indices in shared memory	
fa2pac	convert FASTA to PAC format	
pac2bwt	generate BWT from PAC	
pac2bwtgen	alternative algorithm for generating BWT	
bwtupdate	update .bwt to the new format	
bwt2sa	generate SA from BWT and Occ	

Les principaux outils

Note: To use BWA, you need to first index the genome with ``bwa index'`.

There are three alignment algorithms in BWA: ``mem'`, ``bwasw'`, and ``aln/samse/sampe'`. If you are not sure which to use, try ``bwa mem'` first. Please ``man ./bwa.1'` for the manual.

# Bwa: un mapper puissant, précis et versatile

Usage: `bwa index [options] <in.fasta>`

Options: `-a STR` BWT construction algorithm: `bwtsv`, `is` or `rb2` [`auto`]

`-p STR` prefix of the index [same as fasta name]

`-b INT` block size for the `bwtsv` algorithm (effective with `-a bwtsv`) [10000000]

`-6` index files named as `<in.fasta>.64.*` instead of `<in.fasta>.*`

Warning: ``-a bwtsv'` does not work for short genomes, while ``-a is'` and ``-a div'` do not work not for long genomes.

## Indexations

# Bwa: un mapper puissant, précis et versatile

Usage: `bwa mem [options] <idxbase> <in1.fq> [in2.fq]`

Algorithm options:

`-t INT` number of threads [1]

`-k INT` minimum seed length [19]

`-w INT` band width for banded alignment [100]

`-d INT` off-diagonal X-dropoff [100]

`-r FLOAT` look for internal seeds inside a seed longer than  $\{-k\} * FLOAT$  [1.5]

`-y INT` seed occurrence for the 3rd round seeding [20]

`-c INT` skip seeds with more than INT occurrences [500]

`-D FLOAT` drop chains shorter than FLOAT fraction of the longest overlapping chain [0.50]

`-W INT` discard a chain if seeded bases shorter than INT [0]

`-m INT` perform at most INT rounds of mate rescues for each read [50]

`-S` skip mate rescue

`-P` skip pairing; mate rescue performed unless -S also in use

Local mapping:

`bwa mem`

Algorithm options

Note: Please read the man page for detailed description of the command line and options.

# Bwa: un mapper puissant, précis et versatile

Usage: `bwa mem [options] <idxbase> <in1.fq> [in2.fq]`

Scoring options:

- A INT      score for a sequence match, which scales options -TdBOELU unless overridden [1]
- B INT      penalty for a mismatch [4]
- O INT[,INT] gap open penalties for deletions and insertions [6,6]
- E INT[,INT] gap extension penalty; a gap of size k cost '{-O} + {-E}\*k' [1,1]
- L INT[,INT] penalty for 5'- and 3'-end clipping [5,5]
- U INT      penalty for an unpaired read pair [17]
  
- x STR      read type. Setting -x changes multiple parameters unless overridden [null]
  - pacbio: -k17 -W40 -r10 -A1 -B1 -O1 -E1 -L0 (PacBio reads to ref)
  - ont2d: -k14 -W20 -r10 -A1 -B1 -O1 -E1 -L0 (Oxford Nanopore 2D-reads to ref)
  - intractg: -B9 -O16 -L5 (intra-species contigs to ref)

Local mapping:

`bwa mem`

Scoring options

Note: Please read the man page for detailed description of the command line and options.

# Bwa: un mapper puissant, précis et versatile

Usage: bwa mem [options] <idxbase> <in1.fq> [in2.fq]

Input/output options:

- p smart pairing (ignoring in2.fq)
- R STR read group header line such as '@RG\tID:foo\tSM:bar' [null]
- H STR/FILE insert STR to header if it starts with @; or insert lines in FILE [null]
- j treat ALT contigs as part of the primary assembly (i.e. ignore <idxbase>.alt file)
- v INT verbose level: 1=error, 2=warning, 3=message, 4+=debugging [3]
- T INT minimum score to output [30]
- h INT[,INT] if there are <INT hits with score >80% of the max score, output all in XA [5,200]
- a output all alignments for SE or unpaired PE
- C append FASTA/FASTQ comment to SAM output
- V output the reference FASTA header in the XR tag
- Y use soft clipping for supplementary alignments
- M mark shorter split hits as secondary
- I FLOAT[,FLOAT[,INT[,INT]]]  
specify the mean, standard deviation (10% of the mean if absent), max  
(4 sigma from the mean if absent) and min of the insert size distribution.  
FR orientation only. [inferred]

Local mapping:

bwa mem

In/output options

# Bwa: un mapper puissant, précis et versatile

Usage: `bwa aln [options] <prefix> <in.fq>`

Options: ~~-n NUM max #diff (int) or missing prob under 0.02 err rate (float) [0.04]~~

~~-o INT maximum number or fraction of gap opens [1]~~

-e INT maximum number of gap extensions, -1 for disabling long gaps [-1]

-i INT do not put an indel within INT bp towards the ends [5]

~~-d INT maximum occurrences for extending a long deletion [10]~~

~~-l INT seed length [32]~~

-k INT maximum differences in the seed [2]

-m INT maximum entries in the queue [2000000]

-t INT number of threads [1]

-M INT mismatch penalty [3]

-O INT gap open penalty [11]

-E INT gap extension penalty [4]

-R INT stop searching when there are >INT equally best hits [30]

-q INT quality threshold for read trimming down to 35bp [0]

-f FILE file to write output to instead of stdout

-B INT length of barcode

-L log-scaled gap penalty for long deletions

-N non-iterative mode: search for all n-difference hits (sloooow)

Global mapping:  
`bwa aln`

One type of aDNA option  
`-l 16500 -n 0.01 -o 2`



# Bwa: un mapper puissant, précis et versatile

Usage: bwa samse [-n max\_occ] [-f out.sam] [-r RG\_line] <prefix> <in.sai> <in.fq>

Usage: bwa sampe [options] <prefix> <in1.sai> <in2.sai> <in1.fq> <in2.fq>

Options: -a INT maximum insert size [500]

-o INT maximum occurrences for one end [100000]

-n INT maximum hits to output for paired reads [3]

-N INT maximum hits to output for discordant pairs [10]

-c FLOAT prior of chimeric rate (lower bound) [1.0e-05]

-f FILE sam file to output results to [stdout]

-r STR read group header line such as '@RG\tID:foo\tSM:bar' [null]

-P preload index into memory (for base-space reads only)

-s disable Smith-Waterman for the unmapped mate

-A disable insert size estimate (force -s)

Global mapping:  
bwa aln doit être  
chainé avec bwa  
samse ou sampe

## The alignment

M02279\_0166\_000000000-ALRV1\_BN\_R4278:1:2113:14710:12532 16  
gi|5819095|ref|NC\_001321.1|16398bp|Balaenoptera 2329 0 62M \* 0 0  
AAATTAAAAAAAATAAAGGAACTCGGC AAACACA AACCCCGCCTGTTTACCA AAAAACATCA  
]] XI:Z:GCTCCGT YI:Z:CCCCGG XJ:Z:CCGGTAC YJ:Z:CCCCGG  
FF:i:3 RG:Z:R4237 ZO:i:0 XT:A:R NM:i:2 XO:i:21 X1:i:75 XM:i:2 XO:i:0 XG:i:0 MD:Z:1G10G49

# Output: the sam format

## The alignment

Col	Field	Type	Regex/Range	Brief description
1	QNAME	String	[!-?A-Z]{1,254}	Query template NAME
2	FLAG	Int	$[0, 2^{16} - 1]$	bitwise FLAG
3	RNAME	String	\* [:name:^\*-] [:name:]*	Reference sequence NAME <sup>9</sup>
4	POS	Int	$[0, 2^{31} - 1]$	1-based leftmost mapping POSITION
5	MAPQ	Int	$[0, 2^8 - 1]$	MAPping Quality
6	CIGAR	String	\* ([0-9]+[MIDNSHPX-])+	CIGAR string
7	RNEXT	String	\* [:name:^\*-] [:name:]*	Reference name of the mate/next read
8	PNEXT	Int	$[0, 2^{31} - 1]$	Position of the mate/next read
9	TLEN	Int	$[-2^{31} + 1, 2^{31} - 1]$	observed Template LENGTH
10	SEQ	String	\* [A-Za-z-]+	segment SEQUENCE
11	QUAL	String	[!-~]+	ASCII of Phred-scaled base QUALity+33

# Output: the sam format

## The alignment

FLAG: Combination of bitwise FLAGS.<sup>10</sup> Each bit is explained in the following table:

Bit	Description
1	0x1 template having multiple segments in sequencing
2	0x2 each segment properly aligned according to the aligner
4	0x4 segment unmapped
8	0x8 next segment in the template unmapped
16	0x10 SEQ being reverse complemented
32	0x20 SEQ of the next segment in the template being reverse complemented
64	0x40 the first segment in the template
128	0x80 the last segment in the template
256	0x100 secondary alignment
512	0x200 not passing filters, such as platform/vendor quality controls
1024	0x400 PCR or optical duplicate
2048	0x800 supplementary alignment

# Exemple de commande bwa

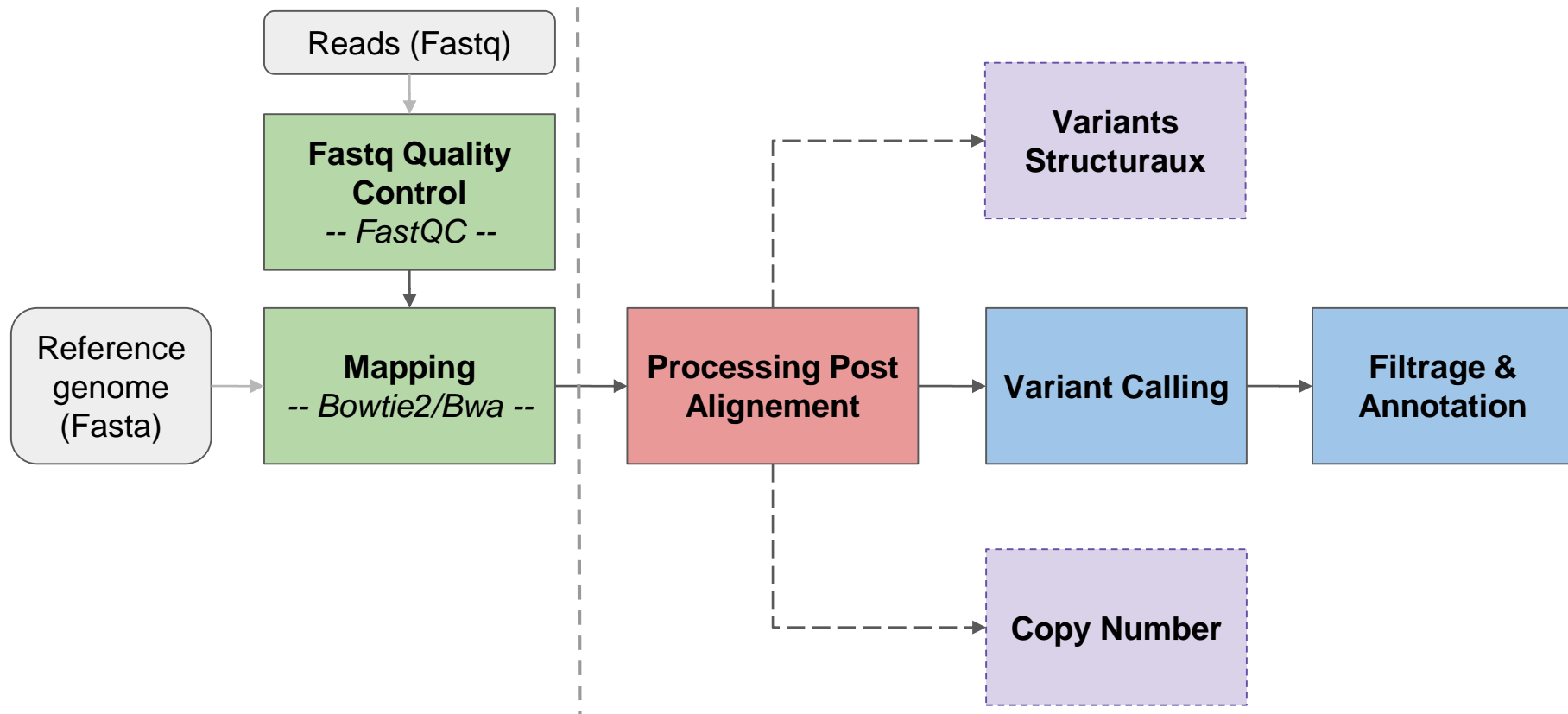
```
srn --mem=8GB --cpus-per-task=4 bwa mem -t 4 genome/Bos_taurus.UMD3.1.dna.toplevel.6.fa \  
sickle/SRR1262731_extract_trim_R1.fq.gz \  
sickle/SRR1262731_extract_trim_R2.fq.gz \  
2> alignment_bwa/SRR1262731_extract_bwa.log.txt \  
| samtools view -hbS | samtools sort > alignment_bwa/SRR1262731_extract.sort.bam
```

# Indexation des alignements

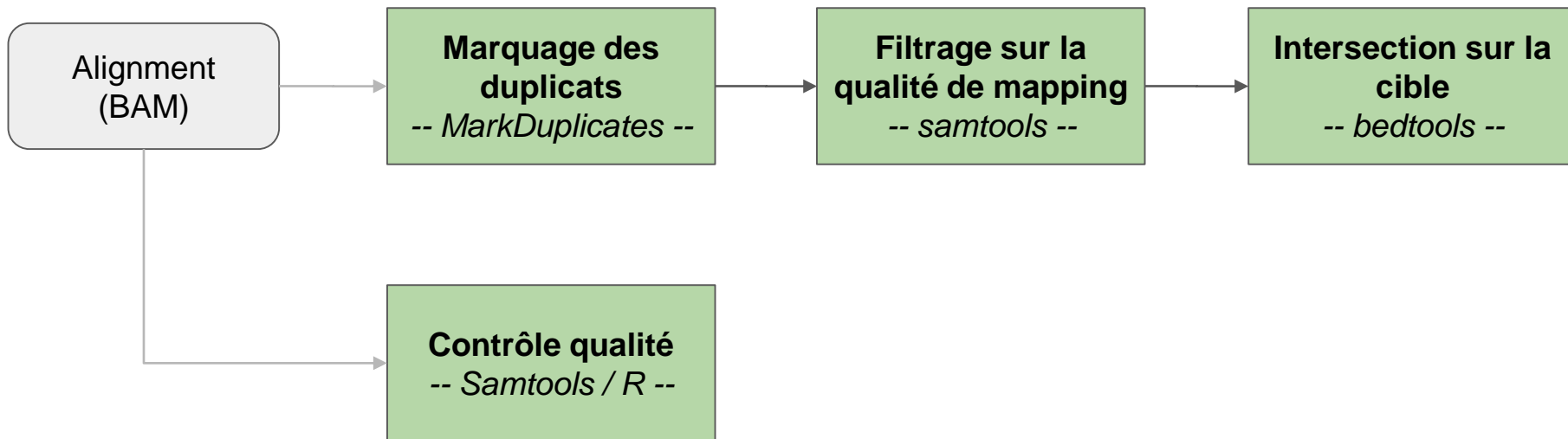
```
srn samtools index alignment_bwa/SRR1262731_extract.sort.bam
```

# Processing Post-Alignment

# Workflow



# Workflow - Processing Post Alignement





# Copie du jeu de données

#Listing des fichiers FASTQ, Genome et BAM

```
$ ls -lh /shared/data/projects/du_bii_2019/data/module5/seance3/fastq
```

```
$ ls -lh /shared/data/projects/du_bii_2019/data/module5/seance3/genome
```

```
$ ls -lh /shared/data/projects/du_bii_2019/data/module5/seance3/alignment_bwa
```

#Se déplacer dans son home

```
$ cd ~
```

#Créer un répertoire de travail

```
$ mkdir M5S3
```

#Copier les données du TP

```
$ cp -r /shared/data/projects/du_bii_2019/data/module5/seance3/* M5S3
```

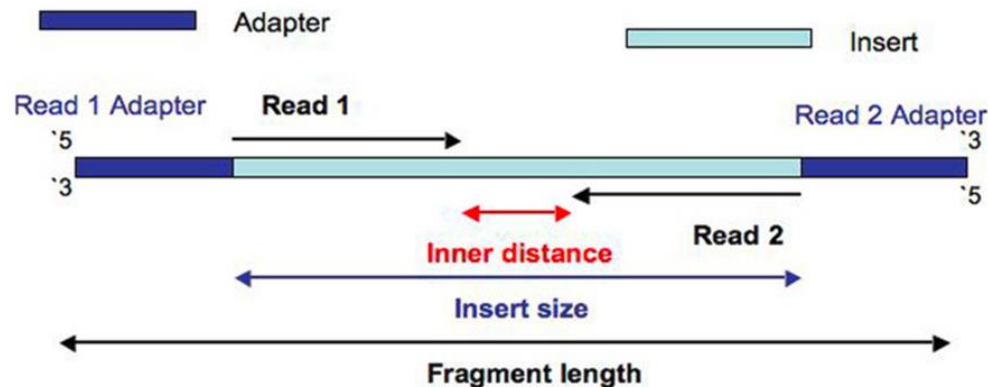
#Se déplacer dans le dossier alignment\_bwa

```
$ cd ~/M5S3/alignment_bwa
```

# Contrôle qualité des données alignées

- Quelles informations regarder une fois le mapping effectué ?
  - Pourcentage total de reads alignés
  - Pourcentage de reads pairés “**proprement**”

- Quels outils ?
  - Samtools flagstat



# Contrôle qualité des données alignées

#Lancement de samtools

\$ **samtools** **--version** # affiche la version (v.1.9)

\$ **samtools** **flagstat** # affiche l'aide

\$ **srun** **samtools** **flagstat** SRR1262731\_extract.sort.bam > SRR1262731.flagstat.txt

\$ **cat** SRR1262731.flagstat.txt # visualisation du résultat

# ReadGroups (RG)

- Associe des informations sur la provenance des reads

→ Identité : run/échantillon

→ Séquençage, librairie...

- Nécessaire à la recherche de variants

```
Mom's data:
@RG      ID:FLOWCELL1.LANE5      PL:ILLUMINA      LB:LIB-MOM-1 SM:MOM
@RG      ID:FLOWCELL1.LANE6      PL:ILLUMINA      LB:LIB-MOM-1 SM:MOM
@RG      ID:FLOWCELL1.LANE7      PL:ILLUMINA      LB:LIB-MOM-2 SM:MOM
@RG      ID:FLOWCELL1.LANE8      PL:ILLUMINA      LB:LIB-MOM-2 SM:MOM

Kid's data:
@RG      ID:FLOWCELL2.LANE1      PL:ILLUMINA      LB:LIB-KID-1 SM:KID
@RG      ID:FLOWCELL2.LANE2      PL:ILLUMINA      LB:LIB-KID-1 SM:KID
@RG      ID:FLOWCELL2.LANE3      PL:ILLUMINA      LB:LIB-KID-2 SM:KID
@RG      ID:FLOWCELL2.LANE4      PL:ILLUMINA      LB:LIB-KID-2 SM:KID
```

- Comment vérifier la présence de ReadGroups dans un fichier BAM?

```
$ samtools view # affiche l'aide
```

```
$ samtools view -H SRR1262731_extract.sort.bam | grep “^@RG”
```

# Comment ajouter des ReadGroups ?

- Au niveau des paramètres du mapper :

Bwa : “ -R @RG\tID:ID\tSM:SAMPLE\_NAME\tPL:Illumina\tPU:PU\tLB:LB”

Bowtie2 : “--rg-id ID --rg SM:SAMPLE\_NAME --rg PL:Illumina --rg PU:PU -  
-rg LB:LB”

- Avec l’outil **AddOrReplaceReadGroups** de la suite PicardTools

```
$ picard AddOrReplaceReadGroups --version # affiche la version (v2.18.9)
```

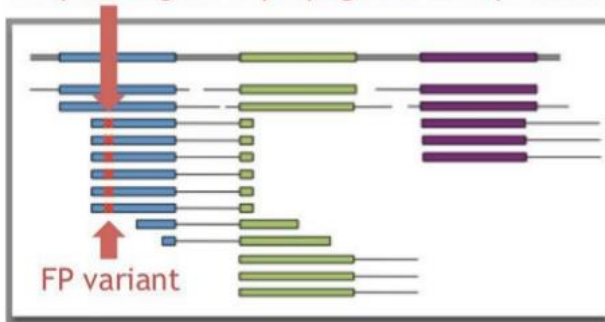
```
$ picard AddOrReplaceReadGroups --help # affiche l'aide
```

```
$ run picard AddOrReplaceReadGroups I=SRR1262731_extract.sort.bam \  
O=SRR1262731_extract.sort.rg.bam RGID=1 RGPL=Illumina RGPU=PU \  
RGSM=SRR1262731 RGLB=LB
```

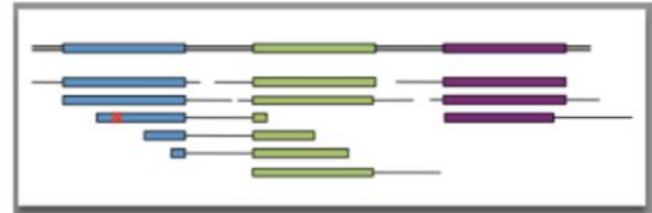
# Marquage des duplicats de PCR

- Identifier les reads provenant d'une même molécule issus de :
  - **PCR duplicates** : amplification PCR durant la préparation de la librairie
  - **Optical duplicates** : cluster illumina identifié comme deux clusters

Sequencing error propagated in duplicates



PCRdup  
removal



# Marquage des duplicats de PCR

- **Garder les duplicats** de PCR : probabilité importante de confondre les duplicats avec des fragments biologiques issus du même locus
- **Marquer les duplicats** mais les conserver dans le fichier BAM : certains outils les supprimeront par défaut (samtools, GATK...)
- **Supprimer les duplicats** du fichier BAM

```
$ picard MarkDuplicates --help # affiche l'aide
$ srun --mem=8GB picard -Xmx8G MarkDuplicates I=SRR1262731_extract.sort.rg.bam \
O=SRR1262731_extract.sort.rg.md.bam M=SRR1262731_extract_metrics_md.txt \
VALIDATION_STRINGENCY=SILENT
$ srun samtools flagstat SRR1262731_extract.sort.rg.md.bam \
> SRR1262731_extract.md.flagstat.txt
$ cat SRR1262731_extract.md.flagstat.txt # nombre de duplicats
$ grep -A1 "LIBRARY" SRR1262731_extract_metrics_md.txt | less -S # % de pcrDup
```

# Filtres sur les alignements

**Restreindre le fichier BAM** en fonction de métriques d'alignements :

- **qualité de mapping** (MAPQ) suffisante
- retrait des reads non mappés

```
# Suppression des reads non mappés et filtre sur les reads avec MAPQ < 30
$ srun samtools view -bh -F 4 -q 30 SRR1262731_extract.sort.rg.md.bam \
> SRR1262731_extract.sort.rg.md.filt.bam

$ srun samtools flagstat SRR1262731_extract.sort.rg.md.filt.bam \
> SRR1262731_extract.filt.flagstat.txt

$ cat SRR1262731_extract.filt.flagstat.txt
```



# Filtres sur les alignements

**Restreindre le fichier BAM en fonction de métriques d'alignements :**

- alignements **intersectant les régions d'intérêt**
- en fonction du nombre de mismatches, de la taille d'insert, de paires mappées sur des chromosomes différents...

```
# Conservation des alignements dans les régions ciblées
$ bedtools --version # affiche la version (v2.27.1)
$ bedtools intersect --help # affiche l'aide

$ srun bedtools intersect -a SRR1262731_extract.sort.rg.md.filt.bam \
-b ../additionnal_data/QTL_BT6.bed \
> SRR1262731_extract.sort.rg.md.filt.onTarget.bam

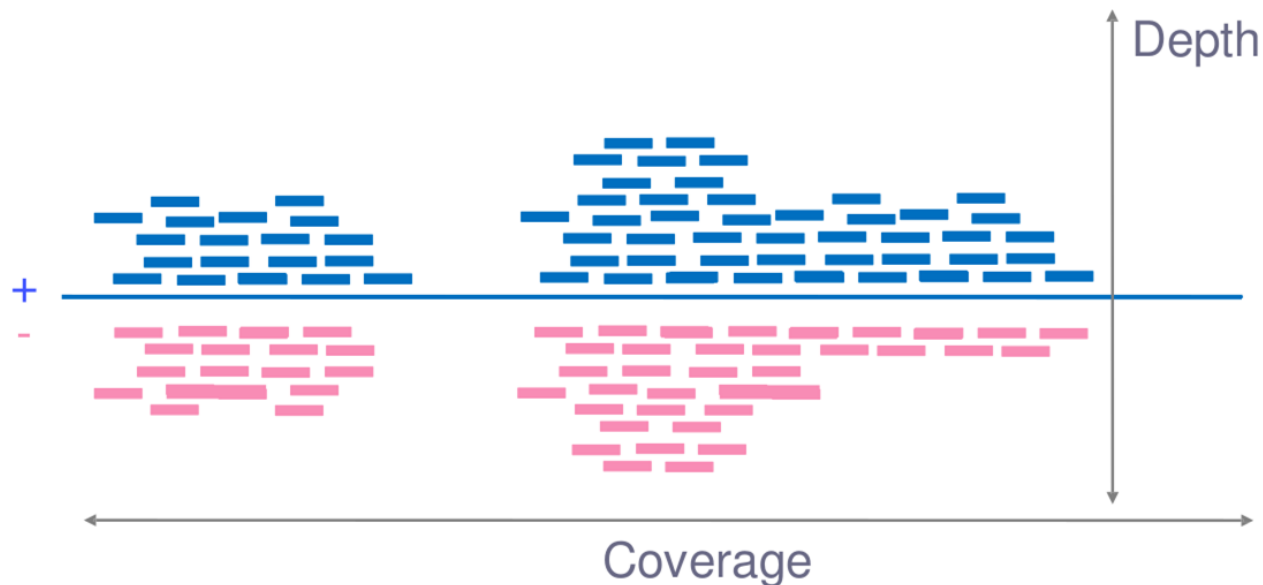
$ srun samtools index SRR1262731_extract.sort.rg.md.filt.onTarget.bam
```

# Analyse de la couverture

Contrôle qualité de l'**enrichissement** de ma capture :

→ Est-ce que ma région est **couverte par suffisamment de reads** ?

→ Cette couverture est-elle homogène sur toute la région ?



# Analyse de la couverture

Contrôle qualité de l'**enrichissement** de ma capture :

→ Est-ce que ma région est **couverte par suffisamment de reads** ?

→ Cette couverture est-elle homogène sur toute la région ?

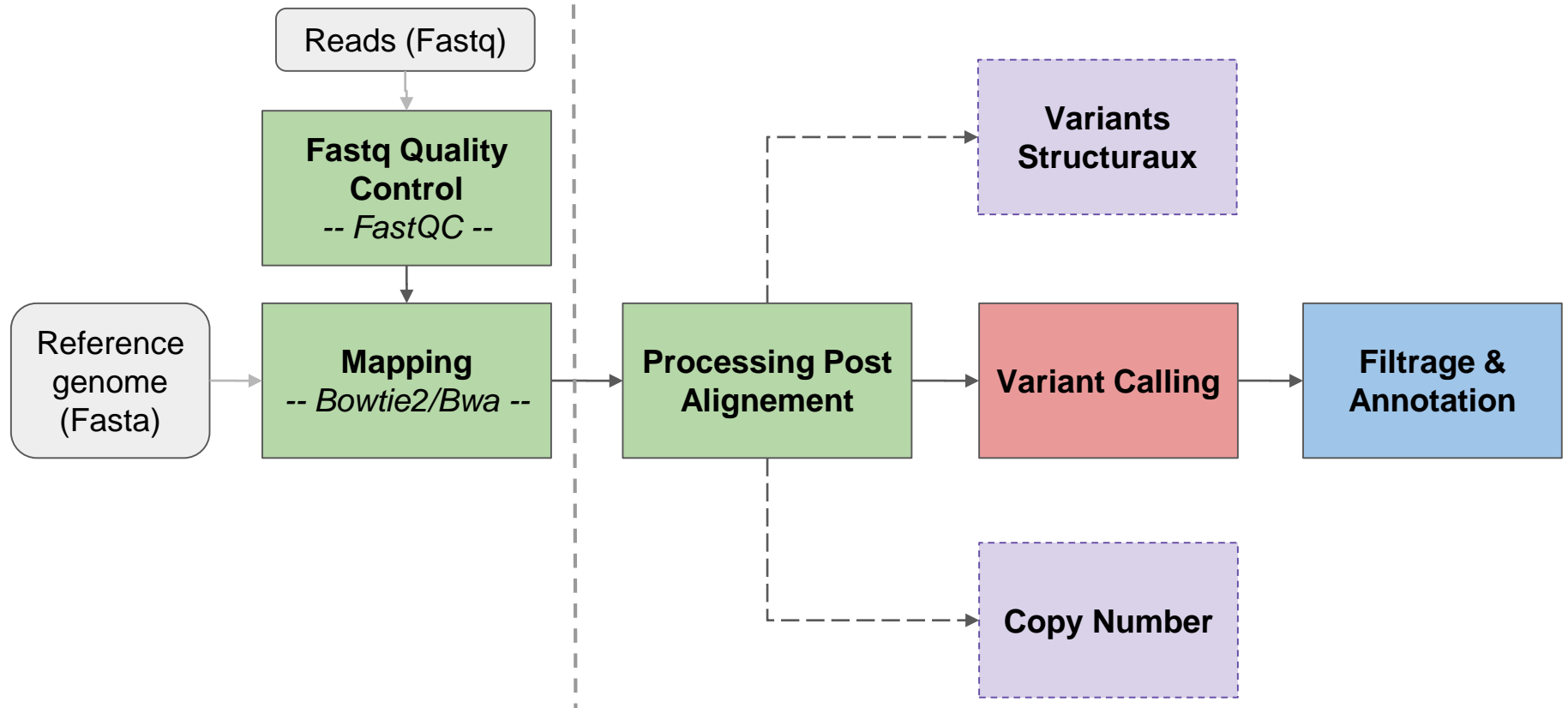
```
# Calcul de la couverture avec samtools
$ samtools depth --help # affiche l'aide

$ srun samtools depth -b ../additionnal_data/QTL_BT6.bed \
SRR1262731_extract.sort.rg.md.filt.onTarget.bam \
> SRR1262731_extract.onTarget.depth.txt

$ head SRR1262731_extract.onTarget.depth.txt
```

# Variant calling

# Workflow



# Définition

**Variant** : variation génomique dans une séquence nucléotidique, en comparaison avec une séquence de référence

- **SNV** : Single Nucleotide Variant
- **INDEL** : INsertion ou DELetion d'une ou plusieurs bases
- **MNV** (Multi-Nucleotide Variant) : plusieurs SNVs et/ou INDELS dans un bloc
- **SV** (Structural Variant) : réarrangement génomique affectant > 50bp

AACGGCC T GTAAC  
AACGGCC A GTAAC

AACGGCC T GTAAC  
AACGGCC - GTAAC

AACGGCC T GTAAC  
AACGGCC AG C GTAAC

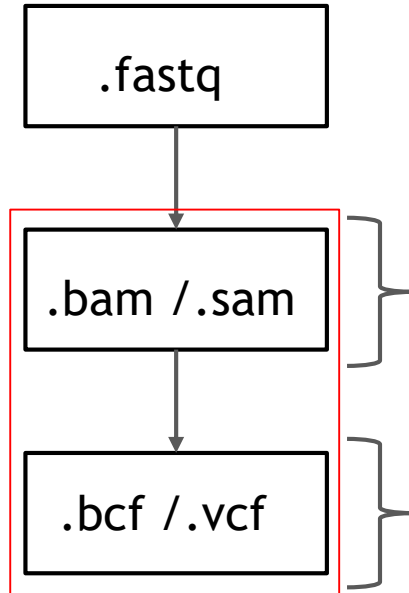
# SNV $\neq$ SNP

- **SNV (Single Nucleotide Variant)**
  - toute altération nucléotidique sans implication de fréquence populationnelle
- **SNP (Single Nucleotide Polymorphism)**
  - implique qu'un variant est partagée dans la population (> 1%)

/!\ l'amalgame SNP est souvent fait pour qualifier les SNVs /!\

# Qu'appelle t-on "Variant Calling"

Détection automatisée des variants (SNVs, Indels de petite taille) à partir d'un fichier contenant des données de séquençage alignées (BAM)

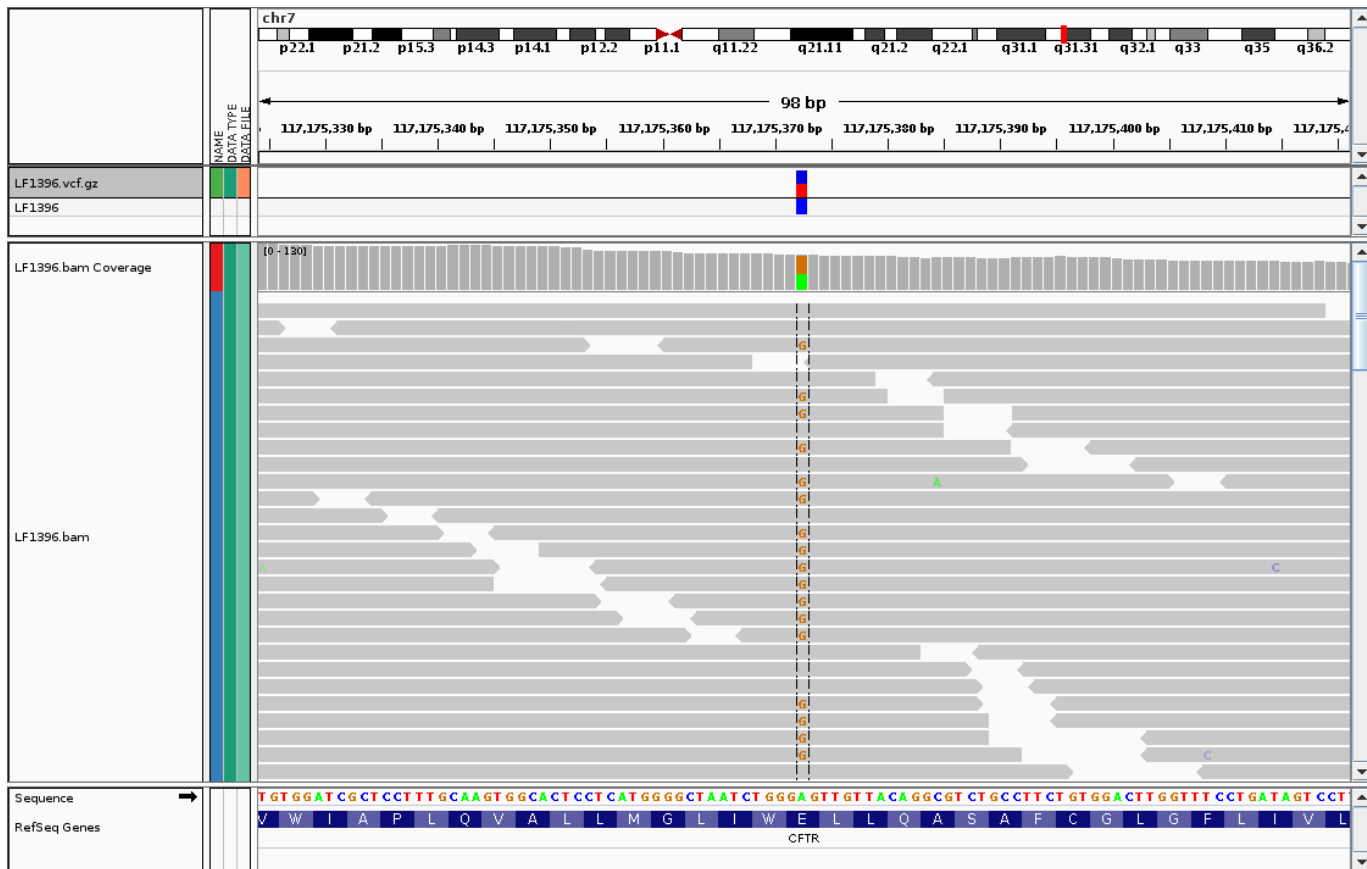


```
HWI-ST1145:74:C101DACXX:7:1102:4284:73714      16      chr20      190930
      CCGTGTTTAAAGGTGGATGCGGTCACCTTCCCAGCTAGGCTTAGGGATTCTTAGTTGGCCTAG
C      BBDCDDCCDDDDDDDDDDDDDDCCCDDBC?DDDDDDDDDDDDDDDDCCDDDDDDDDDDDDCCCCEDD
      AS:i:-15      XM:i:3      X0:i:0      XG:i:0      MD:Z:55C20C13A9      NM:i:3      NH
HWI-ST1145:74:C101DACXX:7:1114:2759:41961      16      chr20      193953
      TGCTGGATCATCTGGTTAGTGGCTTCTGACTCAGAGGACCTTCGTCCCCTGGGGCAGTGGACC
G      DCDDDDDEDDDDDDDDDDDDDDCCDDDDDDDDDEEC>DFFFEJJJJJIGJJJJIHGBHGGJIIJ
```

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	S
chr1	2488153	.	A	G	4476.14	PASS	AC=4;AF=1.00;		
chr1	2491258	.	C	G	2611.42	PASS	AC=2;AF=0.500		



# Qu'appelle t-on "Variant Calling"



# Difficultés - Limitations

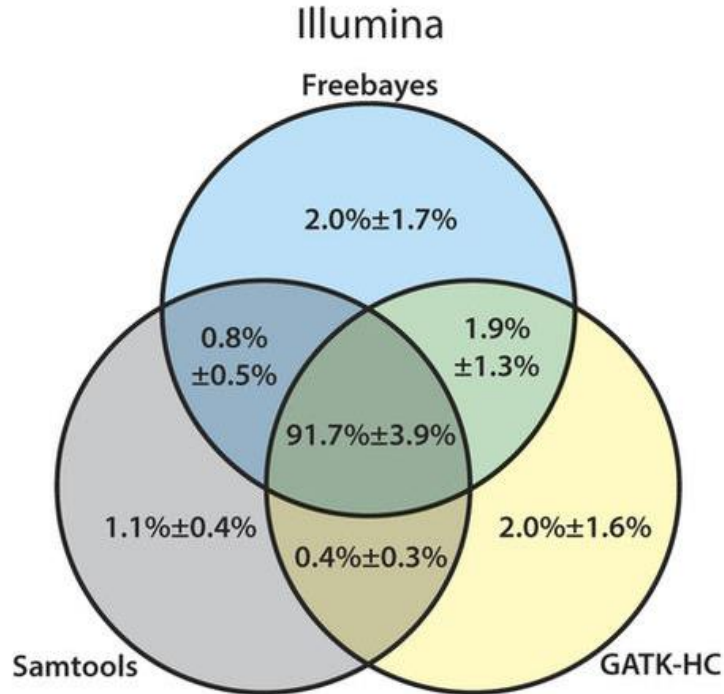
- De nombreux variants **Faux Positifs** peuvent survenir des étapes précédentes :
  - Artéfacts issus des **cycle PCR** pendant la préparation des échantillons
  - Artéfacts issus de l'**amplification en pont** du séquençage (Illumina)
  - **Erreurs de lecture** lors du “BaseCalling”
  - Difficultés d'**alignement** (régions d'ADN répétées)
- Des algorithmes complexes de détection compliquent l'interprétation des résultats

# Variant callers

- Choix du variant caller en fonction de la question biologique
- Utilisés classiquement par la communauté :
  - Samtools mpileup/Bcftools
  - FreeBayes
  - GATK Haplotype Caller
  - Samtools mpileup/VarScan2
  - GATK Mutect (spécifique à la détection tumorale)
  - DiscoSnp (variant calling sans génome de référence)

→ **Aucun outil n'est parfait** : la qualité du calling dépend de l'ensemble du pipeline, des données analysées, et des paramètres utilisés pour filtrer les résultats

# Concordance entre variant callers



- Concordance de **91.7%** entre Freebayes, Samtools, GATK HC (Hwang *et al.*, 2015)
- D'autres analyses montrent des taux plus bas :
  - **70%** (O'Rawe *et al.*, Genome Med, 2013)
  - **57%** (Cornish *et al.*, BioMed, 2015)
- **Rappel (recall) et précision** diffèrent selon les outils et les paramètres utilisés

**/\ Existence de variants qui sont spécifiques aux différents callers /\**

# Recall/Precision

Reference variant set			
		Positive	Negative
Variants Called by the Algorithm	Positive	<b>True Positive (TP)</b> Correct variant allele or position call.	<b>False Positive (FP)</b> Incorrect variant allele or position call.
	Negative	<b>False Negative (FN)</b> Incorrect reference genotype or no call.	<b>True Negative (TN)</b> Correct reference genotype or no call.

## Recall

→ Mesure la capacité de l'outil à détecter le maximum de véritables variants (**sensibilité**)

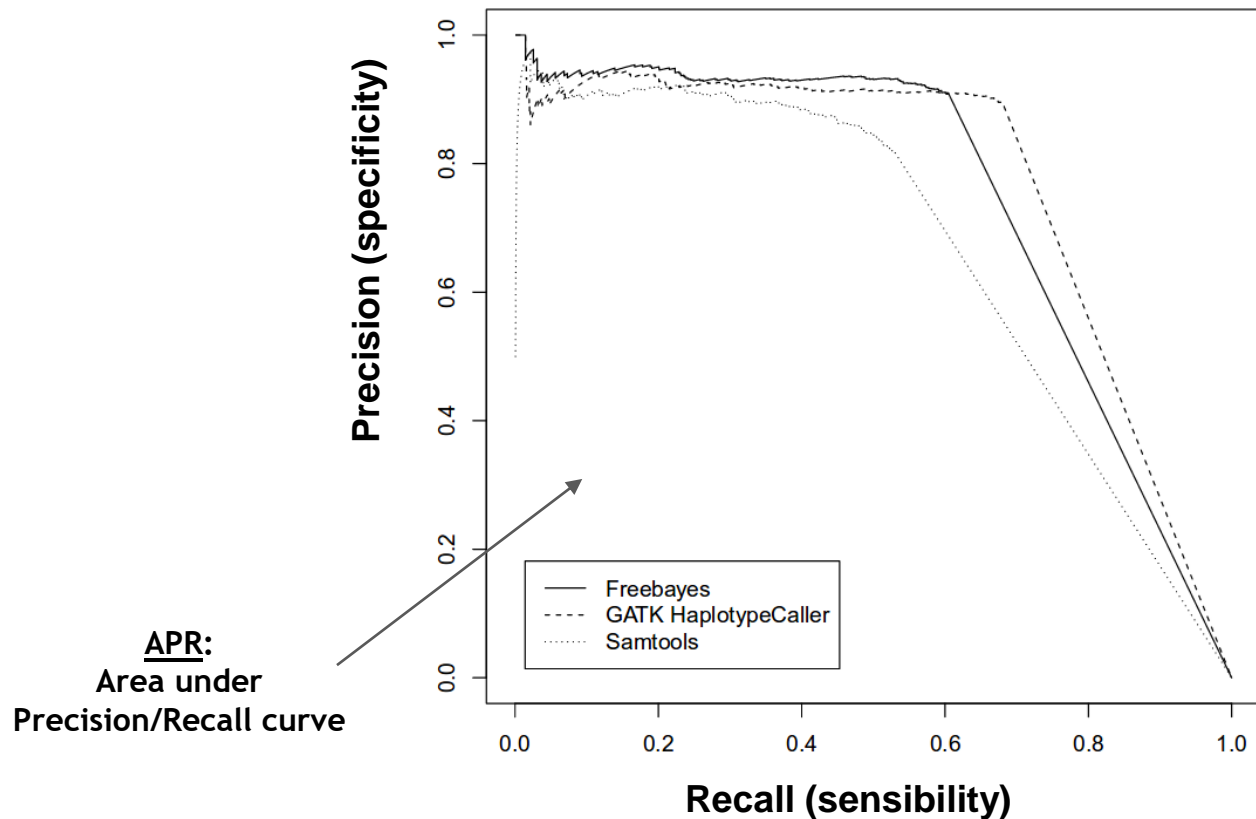
$$\rightarrow TP / (TP + FN)$$

## Precision

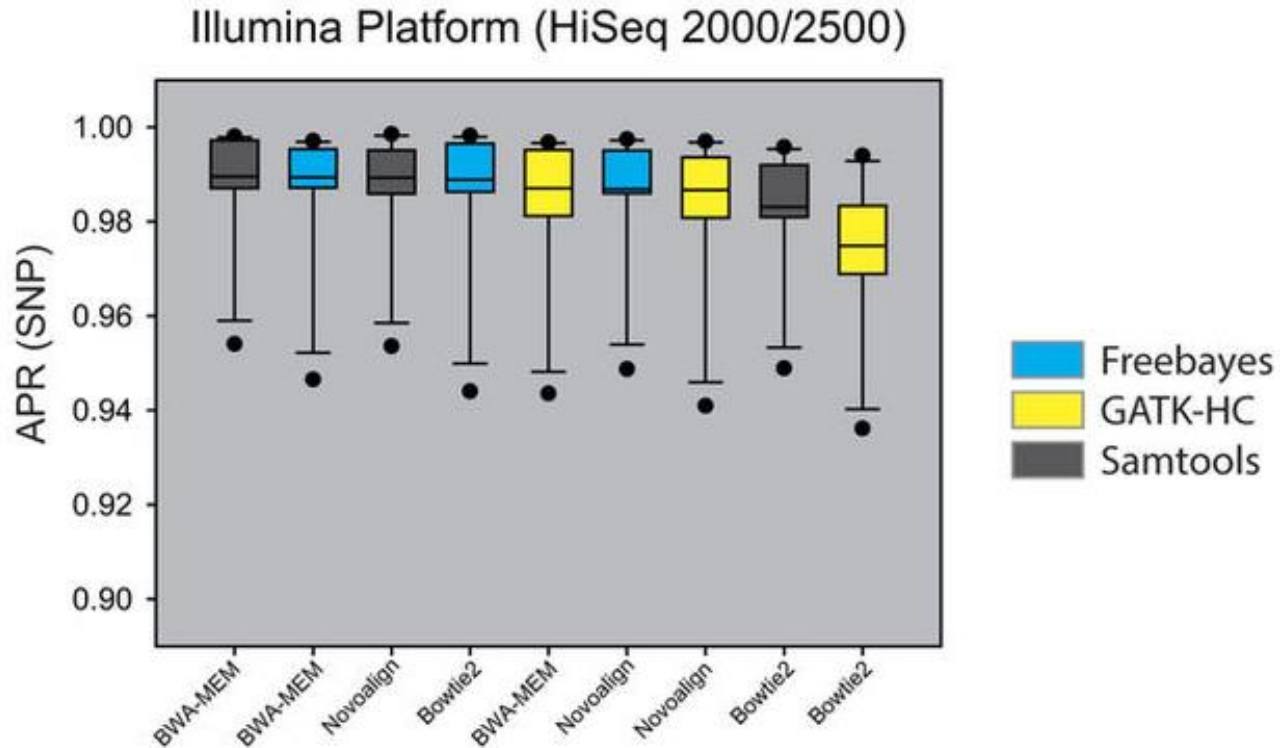
→ Mesure la capacité de l'outil à ne pas détecter de faux variants (**spécificité**)

$$\rightarrow TN / (TN + FP)$$

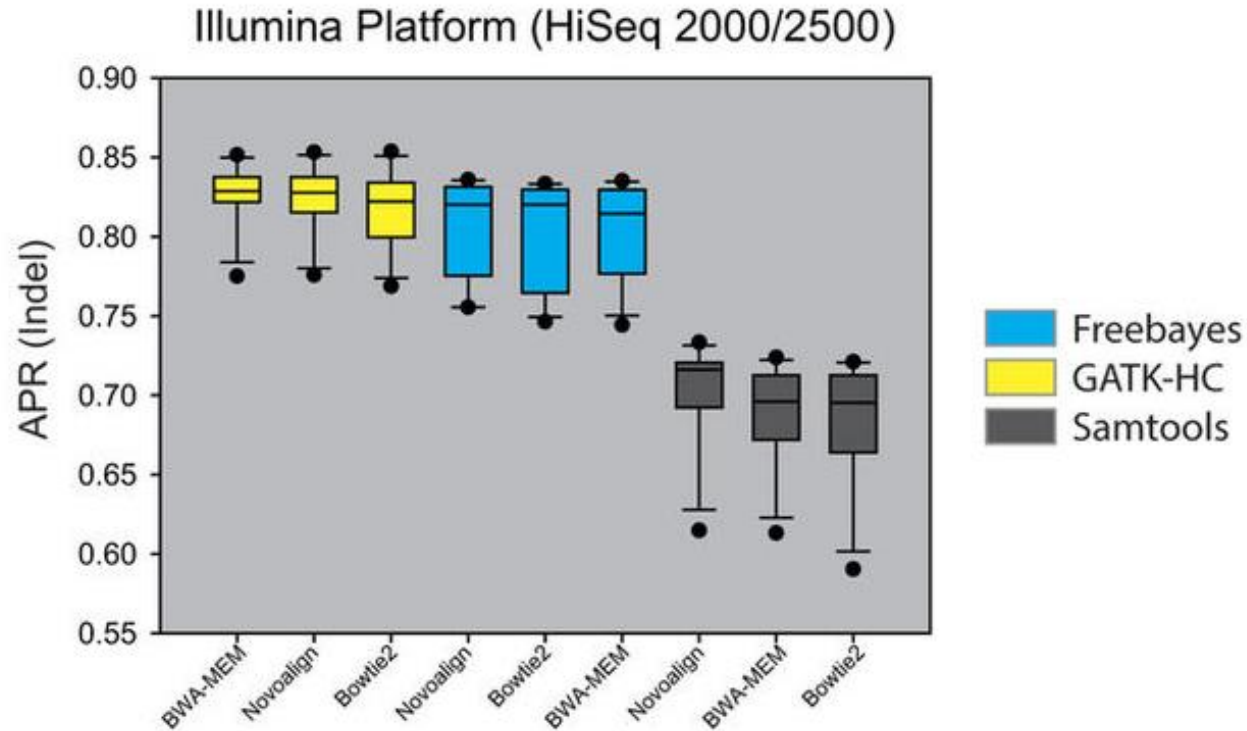
# Recall/Precision



# Performance de la détection de SNVs par l'APR



# Performance de la détection des Indels par l'APR





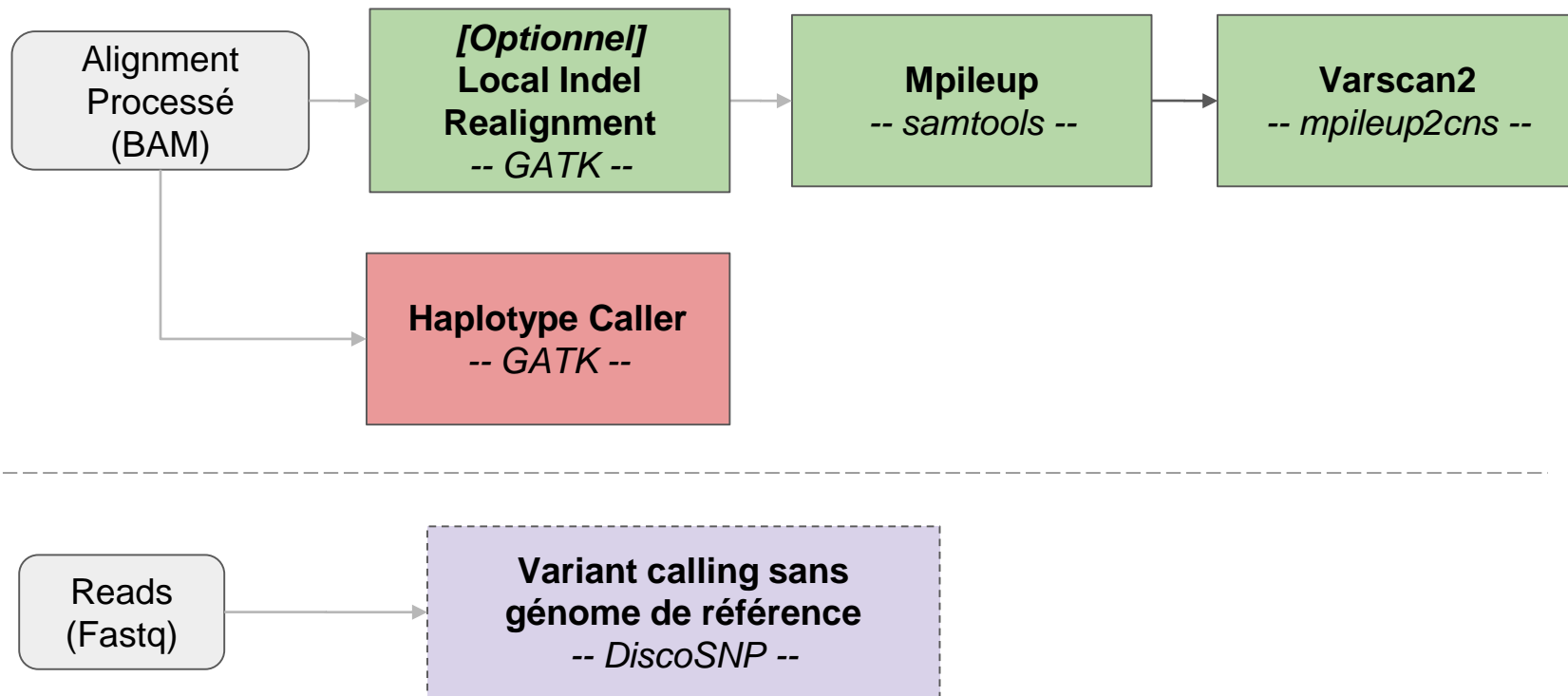
# En conclusion

- La détection de variant permet d'identifier des SNVs et petits Indels à partir d'un fichier d'alignement au format BAM
- De nombreux outils existent pour la détection de variants, leur efficacité dépend de nombreux paramètres (mapping, qualité des données, paramètres de filtrage des résultats)
- Le “recall” et la “precision” permettent d'évaluer la qualité des résultats de détection de variant. Pour un même outil ces mesures varient selon les seuils de qualité utilisés.

# Partie TP

- Utilisation de deux outils : GATK HaplotypeCaller et Varscan2
- **GATK HaplotypeCaller :**
  - GATK (Genome Analysis ToolKit) est une suite d'outils développée par le Broad Institute
  - Bonne documentation (Best Practices)
  - Permet la gestion d'analyse de plusieurs échantillons (format gVCF)
  - Comporte une étape de réalignement local des indel.
  - Algorithme bayésien
- **Varscan2 :**
  - Temps d'exécution plus courts
  - Algorithme basé sur des heuristiques

# Workflow - Variant Calling



# GATK avec sortie VCF

```
$ gatk HaplotypeCaller --version # affiche la version de GATK (v 4.0.10.0)
```

```
$ gatk HaplotypeCaller # affiche l'aide d'HaplotypeCaller
```

Required Arguments:

--input, -I:String            BAM/SAM/CRAM file containing reads    This argument must be specified at least once.

--output, -O:String           File to which variants should be written    Required.

--reference, -R:String        Reference sequence file    Required.

--min-base-quality-score, -mbq:Byte

Minimum base quality required to consider a base for calling

Default value: 10.

# GATK avec sortie VCF

```
$ cd ../  
$ mkdir -p GATK/vcf  
$ cd GATK/
```

```
# Détection de variant GATK avec sortie VCF  
$ srun --mem=8G gatk HaplotypeCaller \  
-I ../alignment_bwa/SRR1262731_extract.sort.rg.md.filt.onTarget.bam \  
-L ../additionnal_data/QTL_BT6.bed \  
-O vcf/SRR1262731_extract_GATK.vcf \  
-R ../genome/Bos_taurus.UMD3.1.dna.toplevel.6.fa \  
--min-base-quality-score 18 \  
--minimum-mapping-quality 30 \  
-ERC "NONE"
```

# GATK avec sortie gVCF

```
# Détection de variants GATK avec sortie gVCF
$ srun --mem=8GB gatk HaplotypeCaller \
-I ../alignment_bwa/SRR1262731_extract.sort.rg.md.filt.onTarget.bam \
-L ../additionnal_data/QTL_BT6.bed -O gvcf/SRR1262731_extract_GATK.g.vcf \
-R ../genome/Bos_taurus.UMD3.1.dna.toplevel.6.fa \
--min-base-quality-score 18 \
--minimum-mapping-quality 30 \
-ERC "GVCF"

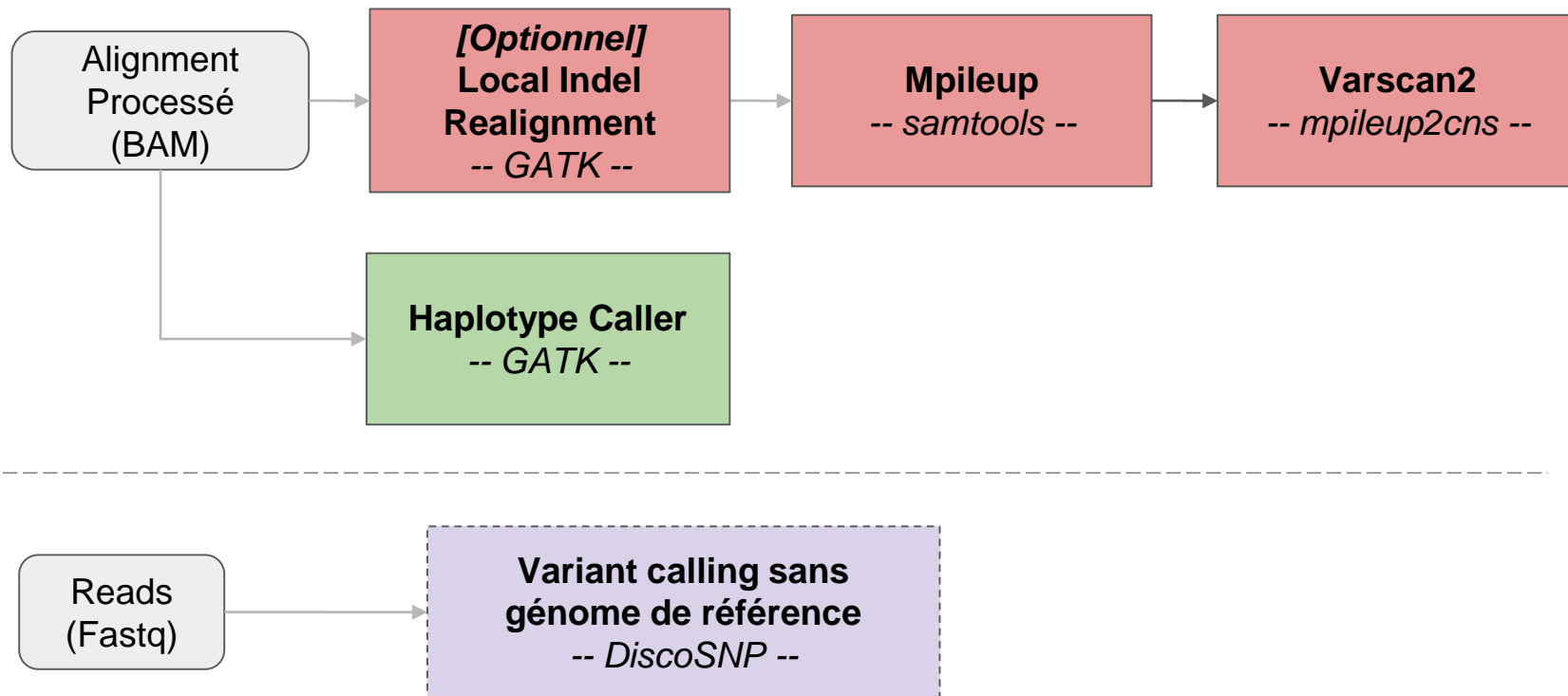
$ ls -ltrh gvcf/
```

# GATK avec sortie gVCF

```
# Fusion des fichier gVCF en un seul gVCF
$ srun --mem=8GB gatk CombineGVCFs \
-L ../additionnal_data/QTL_BT6.bed \
-R ../genome/Bos_taurus.UMD3.1.dna.toplevel.6.fa \
--variant gvcf/SRR1262731_extract_GATK.g.vcf \
--variant gvcf/SRR1205992_extract_GATK.g.vcf \
--variant gvcf/SRR1205973_extract_GATK.g.vcf \
-O gvcf/pool_GATK.g.vcf
```

```
# Détection de variants simultanée sur les 3 échantillons du gVCF
$ srun --mem=8GB gatk GenotypeGVCFs -R
../genome/Bos_taurus.UMD3.1.dna.toplevel.6.fa \
--variant gvcf/pool_GATK.g.vcf \
-O gvcf/pool_GATK.vcf
```

# Workflow - Variant Calling





# Samtools mpileup/Varscan2

```
# Affichage de l'aide de samtools mpileup
```

```
$ samtools mpileup
```

```
Usage: samtools mpileup [options] in1.bam [in2.bam [...]]
```

```
-q, --min-MQ INT          skip alignments with mapQ smaller than INT [0]
```

```
# L'aide de Varscan s'affiche avec le lancement de $ varscan (v2.4.3)
```

```
# Affichage de l'aide de varscan mpileup2cns
```

```
$ varscan mpileup2cns -h
```

```
USAGE: java -jar VarScan.jar mpileup2cns [pileup file] OPTIONS
```

```
mpileup file - The SAMtools mpileup file
```

```
OPTIONS:
```

```
--min-coverage      Minimum read depth at a position to make a call [8]
```

```
--min-reads2       Minimum supporting reads at a position to call variants [2]
```

```
--min-avg-qual      Minimum base quality at a position to count a read [15]
```

# Samtools mpileup/Varscan2

```
# Creation d'un nouveau dossier
```

```
$ cd ..
```

```
$ mkdir -p Varscan
```

```
$ cd Varscan
```

```
# Conversion du fichier d'alignement "bam" en format "mpileup"
```

```
$ srun samtools mpileup -q 30 -B -d 10000 -f
```

```
../genome/Bos_taurus.UMD3.1.dna.toplevel.6.fa \
```

```
../alignment_bwa/SRR1262731_extract.sort.rg.md.filt.onTarget.bam \
```

```
> SRR1262731_extract.mpileup # -A pour garder les paires anormales
```

```
# Détection de variants avec Varscan
```

```
$ srun varscan mpileup2cns SRR1262731_extract.mpileup --output-vcf --variants --  
min-avg-qual 18 > SRR1262731_extract_Varscan.vcf
```

# Vcf-merge

```
$ bgzip # v1.9
$ tabix # v1.9
$ vcftools # v0.1.16

# Renommer l'échantillon dans le VCF
$ sed -i 's|Sample1|SRR1262731.Varscan|g' SRR1262731_extract_Varscan.vcf

# Compression et indexation du fichiers vcf
$ bgzip -c SRR1262731_extract_Varscan.vcf > SRR1262731_extract_Varscan.vcf.gz
$ tabix -p vcf SRR1262731_extract_Varscan.vcf.gz

# Merge des trois échantillons appelés avec Varscan
$ srun vcf-merge SRR1262731_extract_Varscan.vcf.gz
SRR1205992_extract_Varscan.vcf.gz SRR1205973_extract_Varscan.vcf.gz >
pool_Varscan.vcf
```

# Le format VCF

<http://genome.jouy.inra.fr/~orue/VCF/vcf.html>

# Reading a multi-samples VCF

**VCF header**

```
##fileformat=VCFv4.0
##fileDate=20100707
##source=VCFtools
##reference=NCBI36
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality (phred score)">
##FORMAT=<ID=GL,Number=3,Type=Float,Description="Likelihoods for RR,RA,AA genotypes (R=ref,A=alt)">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##ALT=<ID=DEL,Description="Deletion">
##INFO=<ID=SVTYPE,Number=1,Type=String,Description="Type of structural variant">
##INFO=<ID=END,Number=1,Type=Integer,Description="End position of the variant">
```

**Mandatory header lines**

**Optional header lines** (meta-data about the annotations in the VCF body)

**Body**

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	SAMPLE1	SAMPLE2
1	1	.	ACG	A,AT	.	PASS	.	GT:DP	1/2:13	0/0:29
1	2	rs1	C	T,CT	.	PASS	H2;AA=T	GT:GQ	0/1:100	2/2:70
1	5	.	A	G	.	PASS	.	GT:GQ	1/0:77	1/1:95
1	100	.	T	<DEL>	.	PASS	SVTYPE=DEL;END=300	GT:GQ:DP	1/1:12:3	0/0:20

**Reference alleles (GT=0)**

**Alternate alleles (GT>0 is an index to the ALT column)**

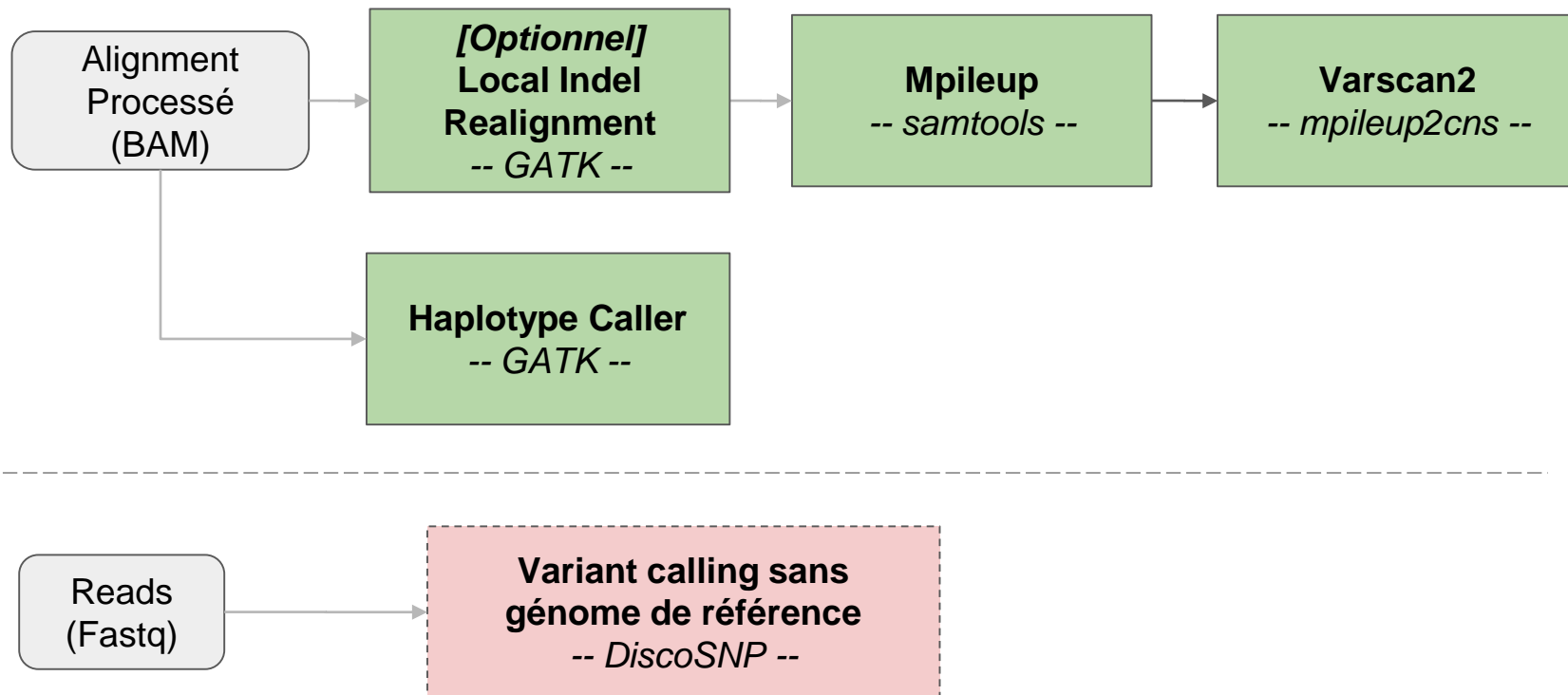
**Deletion**

**SNP**

**Insertion**

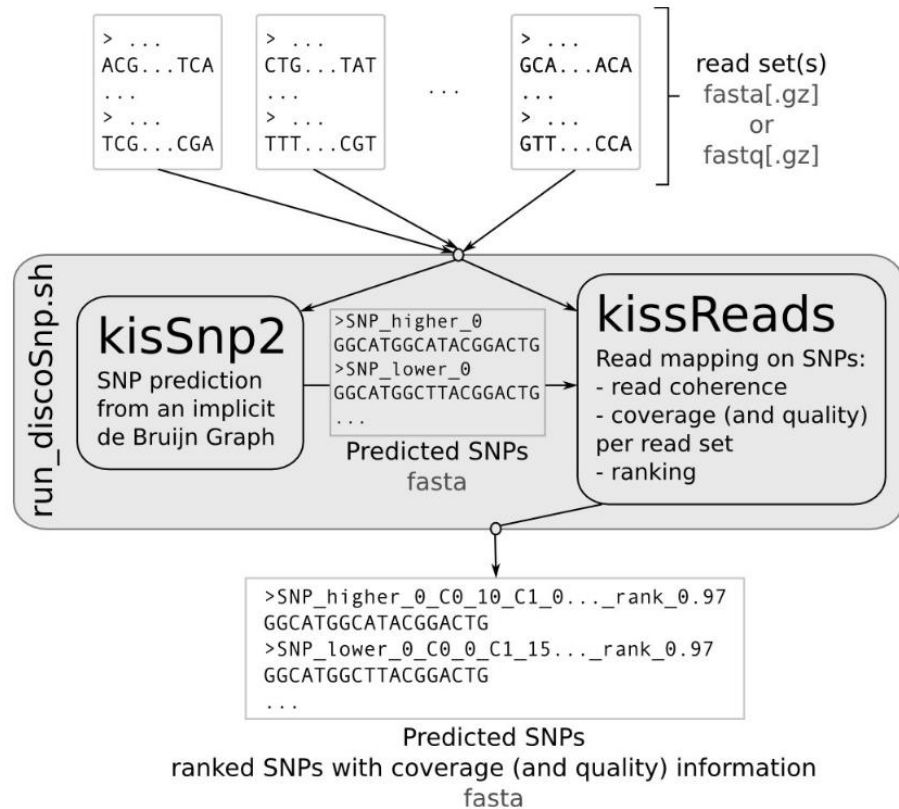
**Other event**

# Workflow - Variant Calling



# Variant Calling sans génome de référence

**Discosnp++** : discovering  
Single Nucleotide Polymorphism  
(SNP) and Indels from raw  
set(s) of reads



# Méthode probalistique VS heuristique

- **Méthode heuristique**

- utilise des seuils pour valider ou non les variants (fréquence allélique, couverture en read, score de qualité)

- **Méthode probabilistique (modèle Bayésien)**

- utilise des modèles statistiques pour estimer la probabilité de chaque génotype possible, en prenant en compte les différents biais pouvant introduire du bruit dans les données



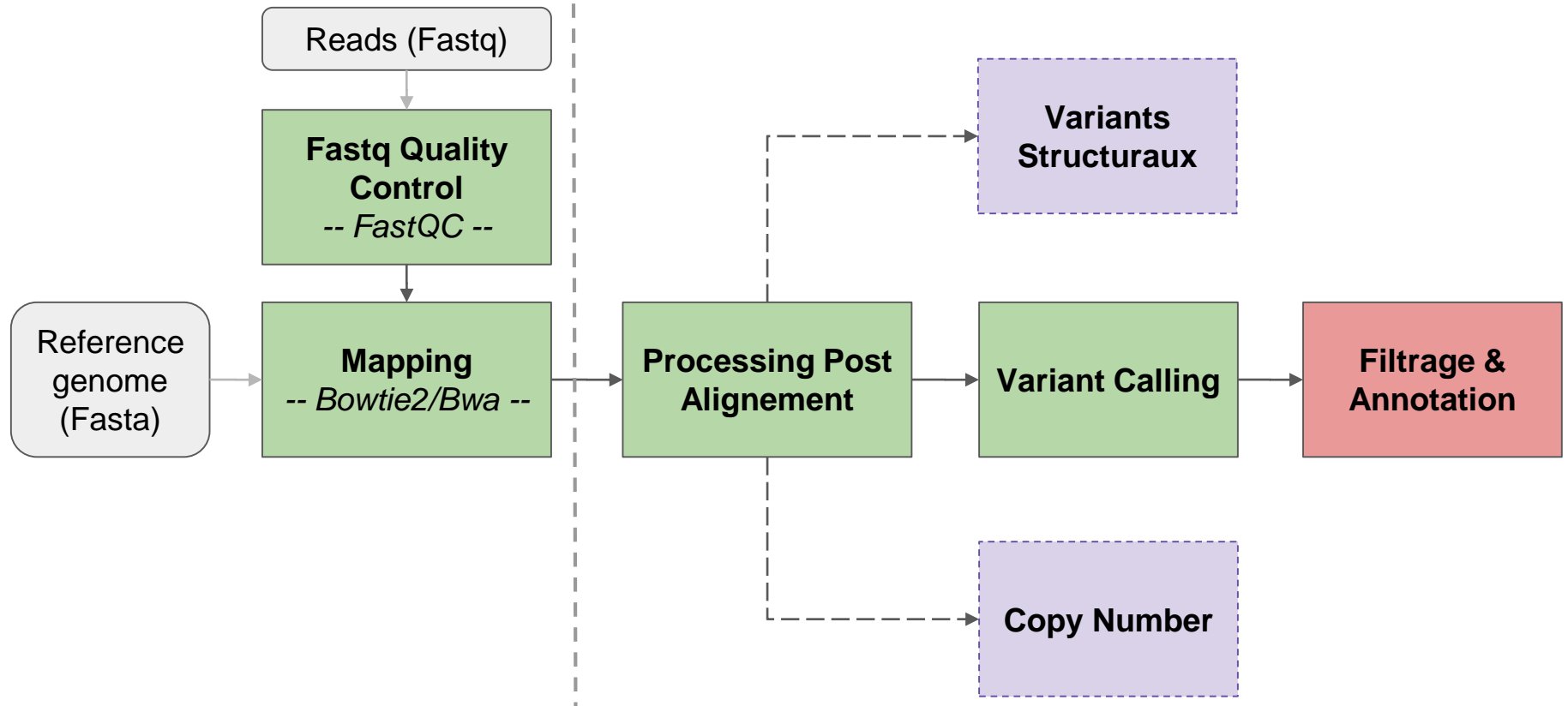
# Exemple de commande Freebayes

# Détection de variants avec Freebayes, -C correspond au nombre de reads minimum supportant le variant pour le reporter

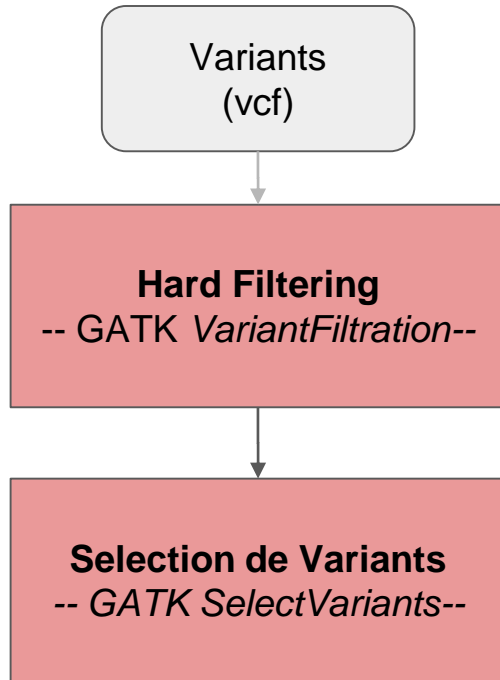
```
$ freebayes -f ../genome/Bos_taurus.UMD3.1.dna.toplevel.6.fa -C 10 \  
../alignment_bwa/SRR1262731_extract.sort.rg.md.filt.onTarget.bam \  
> SRR1262731_freebayes.vcf
```

# Filterage & Annotation

# Workflow



# Workflow - Filtrage et Annotation



# Filtres des variants

- De **nombreux filtres** peuvent être appliqués sur le VCF
  - type de variants à garder (SNVs seulement, Indels...)
  - région d'intérêt
  - seuils arbitraires : profondeur, génotype (0/1, 1/1), ratio allélique...
- Filtres difficilement transposables entre analyse :
  - dépendent de la **question biologique**
  - dépendent des outils utilisés
- **GATK Bests Practices** : recommandations selon des métriques spécifiques à GATK, différentes pour les SNVs des Indels

# SelectVariants et Hard filtering

```
# Préparation d'un nouveau répertoire de résultats
```

```
$ cd ..  
$ mkdir filter_and_annot  
$ cd filter_and_annot
```

```
# Extraction des SNVs dans un fichier séparé pour GATK
```

```
$ srun --mem=8GB gatk SelectVariants -R ../genome/Bos_taurus.UMD3.1.dna.toplevel.6.fa  
\   
-V ../GATK/gvcf/pool_GATK.vcf \  
-O pool_GATK.SNP.vcf \  
--select-type SNP
```

```
# Extraction des SNVs dans un fichier séparé pour VarScan
```

```
$ srun --mem=8GB gatk SelectVariants -R ../genome/Bos_taurus.UMD3.1.dna.toplevel.6.fa  
\   
-V ../VarScan/pool_VarScan.vcf \  
-O pool_VarScan.SNP.vcf \  
--select-type SNP
```

# SelectVariants et Hard filtering

- **QD** - QualByDepth : Score QUAL / AD [profondeur allélique]
- **FS** - FisherStrand : Score estimant un éventuel biais de brin
- **MQ** - MappingQuality : Qualité de mapping moyenne sur l'ensemble du read
- **MQRankSum** : Teste un biais de différence de qualité de mapping entre allèles
- **ReadPosRankSum** : Teste un biais de position des allèles le long du read

[HowTo: Apply hard filters to a call set](#)

doc GATK

[Understanding and adapting the generic hard-filtering recommendations](#)

# SelectVariants et Hard filtering

```
# Filtrage des SNVs selon les filtres recommandés par GATK
```

```
$ srun --mem=8GB gatk VariantFiltration -R  
../genome/Bos_taurus.UMD3.1.dna.toplevel.6.fa \  
-V pool_GATK.SNP.vcf \  
-O pool_GATK.SNP.prefilt.vcf \  
--filter-expression "QD < 2.0 || FS > 60.0 || MQ < 40.0 || MQRankSum < -12.5 ||  
ReadPosRankSum < -8.0" \  
--filter-name "hard_filtering_snv"
```

```
# Sélection des variants passant ce filtre
```

```
$ srun --mem=8GB gatk SelectVariants -R ../genome/Bos_taurus.UMD3.1.dna.toplevel.6.fa  
\  
-V pool_GATK.SNP.prefilt.vcf \  
-O pool_GATK.SNP.filtered.vcf \  
--exclude-filtered
```



# Intersection des résultats des variant callers

```
# Intersection des variants obtenus avec Varscan et avec GATK
```

```
$ vcftools # v0.1.16
```

```
# Compression et indexation des fichiers vcfs
```

```
$ srun bgzip -c pool_GATK.SNP.filtered.vcf > pool_GATK.SNP.filtered.vcf.gz
```

```
$ srun tabix -p vcf pool_GATK.SNP.filtered.vcf.gz
```

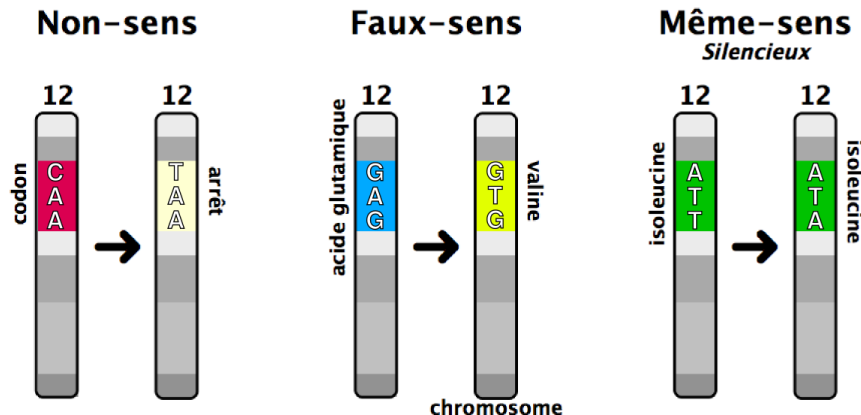
```
$ srun bgzip -c pool_Varscan.SNP.vcf > pool_Varscan.SNP.vcf.gz
```

```
$ srun tabix -p vcf pool_Varscan.SNP.vcf.gz
```

```
$ srun vcf-isec -f -n +2 pool_GATK.SNP.filtered.vcf.gz pool_Varscan.SNP.vcf.gz >  
GATK_varscan_inter.vcf
```

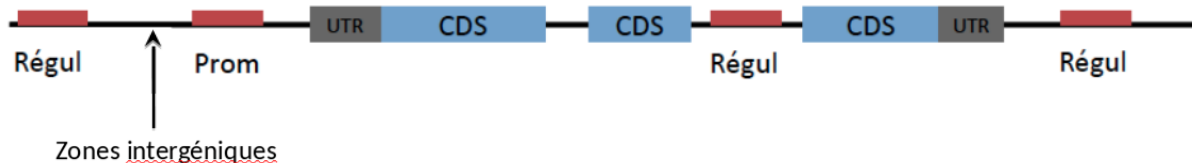
# Annotation des variants


- Ajout d'**informations biologiques pertinentes** aux variants :
  - Est-ce que mes variants sont connus ?
  - Où se positionnent mes variants ?
  - Quel est l'effet d'une mutation sur le CDS qui le contient ?





# Annotation des variants

- Annotation structurale :  
→ Mon variant se trouve-t-il dans un **intron**, un **exon** ?
- Annotation fonctionnelle :  
→ Informations sur la région ? Exemple : CDS codant pour une protéine
- Impacts potentiels :  
→ Dans le cas d'un CDS, **protéine produite tronquée**, allongée, décalée... ou silencieuse (redondance du code génétique)



 Sites de régulation

 CDS Coding Dna Sequence

 UTR Untranslated regions 85

# Annotation des variants

- Nécessité d'avoir des **bases de données** associées aux organismes étudiés (Ensembl, Refseq...)
- Exemples d'outils/algorithmes :
  - SnpEff
  - VEP
  - Annovar
  - SIFT, POLYPHEN2, CADD...

# Snpeff

```
# Création de la base de données Snpeff
$ snpeff -version # affiche la version (v4.3t)

$ echo BosTaurus.genome >> snpeff.config # <genome_name>.genome
$ mkdir -p BosTaurus
$ cp ../genome/Bos_taurus.UMD3.1.dna.toplevel.6.fa BosTaurus/sequences.fa
$ cp ../genome/Bos_taurus.UMD3.1.93_6.gtf BosTaurus/genes.gtf
$ echo -e "BosTaurus\nSnpeff4.1" > BosTaurus.db

$ srun snpeff build -c snpeff.config -gtf22 -v BosTaurus -dataDir .

# Annotation avec notre base de données
$ srun snpeff eff -c snpeff.config -dataDir . BosTaurus -s snpeff_resultat.html
GATK_varscan_inter.vcf > GATK_varscan_inter.annot.vcf
```

# SnpSift

```
$ SnpSift filter -h # affiche l'aide (v 4.3t)
```

```
# Garder les variants codant qui ne sont pas des synonymes :
```

```
$ srun cat GATK_varscan_inter.annot.vcf | SnpSift filter "(ANN[*].EFFECT !=  
'synonymous_variant') && (ANN[*].BIOTYPE = 'protein_coding')" >  
GATK_varscan_inter.annot.coding.nosyn.vcf
```

```
# Sélectionner notre variant d'intérêt parmi les variants hétérozygotes ayant un  
impact (missense)
```

```
$ srun cat GATK_varscan_inter.annot.coding.nosyn.vcf | SnpSift filter  
"ANN[*].EFFECT = 'missense_variant' & isHet( GEN[2] ) & isVariant( GEN[2] ) &  
isRef( GEN[0] ) & isRef( GEN[1] )" >  
GATK_varscan_inter.annot.coding.nosyn.filtered.vcf
```

# Variant d'intérêt

- Quelle type de mutation est impliquée dans notre phénotype d'intérêt pour l'individu SRR1262731 ?
- Quel est son génotype ? Sur quel gène se situe-elle ?
- Qu'en est-il pour les autres individus ?

→ Le variant est **hétérozygote ALT (0/1)** pour l'individu SRR1262731, il comporte une mutation de type SNP (A → C) située sur le gène **ABCG2**, en position **38027010** du **chromosome 6**.

→ Pour les deux autres individus, il ne comporte pas cette mutation : il est homozygote référence (GT: 0/0).