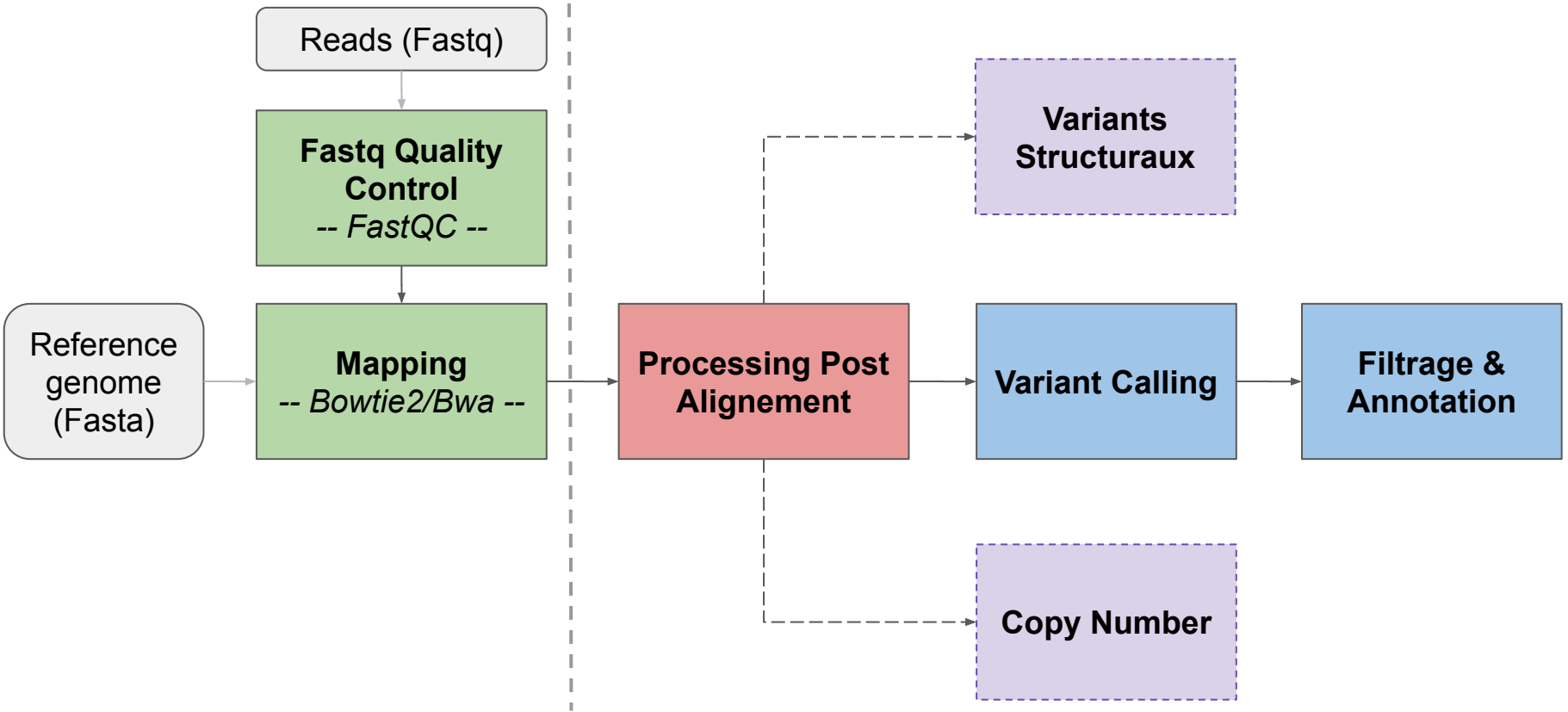
Processing Post-Alignement

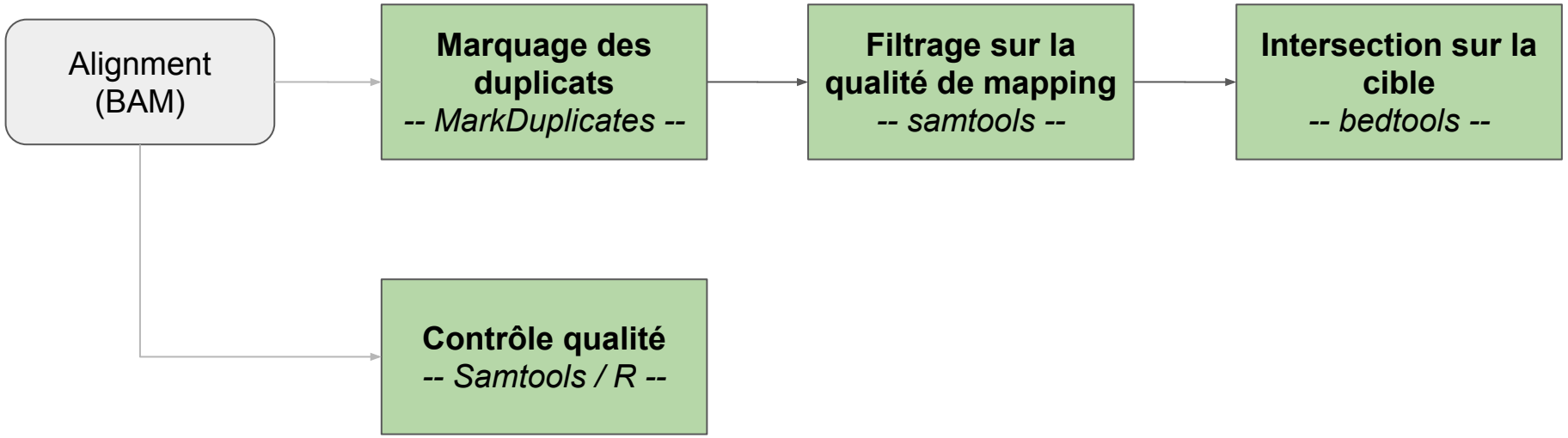
Olivier Rué - INRAE

École de bioinformatique AVIESAN-IFB-INSERM 2020

Workflow


Reads (Fastq)
Fastq Quality Control
-- FastQC --
Reference genome (Fasta)
Mapping
-- Bowtie2/Bwa --
Processing Post Alignement
Variants Structuraux
Variant Calling
Filtrage & Annotation
Copy Number

Workflow - Processing Post Alignement



Copie du jeu de données #1

```
# Listing des fichiers FASTQ, Genome et BAM
$ ls -lh /shared/projects/ebaii2020/atelier_variant/data/variants/fastq
$ ls -lh /shared/projects/ebaii2020/atelier_variant/data/variants/genome
$ ls -lh /shared/projects/ebaii2020/atelier_variant/data/variants/alignment_bwa
```

```
# Copie des fichiers dans notre home
$ mkdir -p ~/tp_variant
$ cp -r /shared/projects/ebaii2020/atelier_variant/data/variants/* ~/tp_variant

# Se déplacer dans le dossier alignment_bwa
$ cd ~/tp_variant/alignment_bwa

# Créer un dossier logs pour stocker les logs (!) des “jobs” SLURM
$ mkdir logs
```

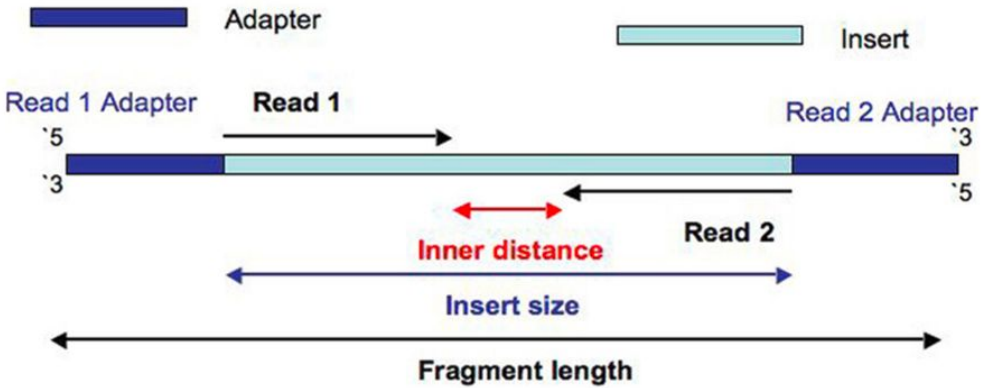
Contrôle qualité des données alignées

- Quelles informations regarder une fois le mapping effectué ?

  → Pourcentage total de reads alignés

  → Pourcentage de reads pairés “**proprement**”

- Quels outils ?
  - Samtools flagstat
  - Qualimap [optionnel]



Contrôle qualité des données alignées

```
# Lancement de samtools
$ module load samtools/1.10
$ samtools --version              # affiche la version (v.1.10)
$ samtools flagstat               # affiche l'aide

$ sbatch -J flagstat1 -o logs/flagstat1.out -e logs/flagstat1.err --wrap=" \
samtools flagstat SRR1262731_extract.sort.bam > SRR1262731.flagstat.txt"

$ cat SRR1262731.flagstat.txt     # visualisation du résultat
```

Contrôle qualité des données alignées

```
# Lancement de Qualimap
$ module load qualimap/2.2.2b
$ qualimap --version              # affiche la version (v2.2.2)
$ qualimap bamqc                  # affiche l'aide

$ sbatch -J qualimap -o logs/qualimap.out -e logs/qualimap.err --wrap=" \
    unset DISPLAY; \
    qualimap bamqc -nt 4 -outdir SRR1262731_extract_qualimap_report \
    --java-mem-size=4G -bam SRR1262731_extract.sort.bam"

# Visualisation du html de sortie en passant par MobaXterm/Cyberduck
```

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"

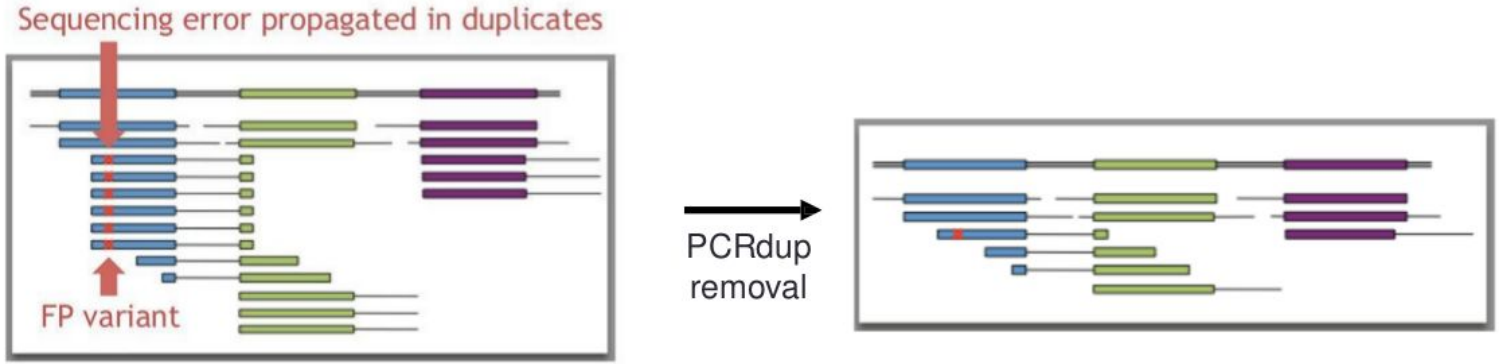
Bowtie2 : "--rg-id ID --rg SM:SAMPLE_NAME --rg PL:Illumina --rg PU:PU --rg LB:LB"

- Avec l'outil **AddOrReplaceReadGroups** de la suite **PicardTools** intégrée à la suite **GATK4**

```
$ module load gatk4/4.1.7.0
$ gatk AddOrReplaceReadGroups --version    # affiche la version (Picard v2.18.9)
$ gatk AddOrReplaceReadGroups --help       # affiche l'aide

$ sbatch -J addRG -o logs/addRG.out -e logs/addRG.err --wrap=" \
    gatk AddOrReplaceReadGroups -I SRR1262731_extract.sort.bam \
    --RGID 1 --RGPL Illumina --RGPU PU --RGSM SRR1262731 --RGLB LB \
    -O SRR1262731_extract.sort.rg.bam"
```

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



Marquage des duplicats de PCR

→ Garder les duplicats : 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…) ou Supprimer les duplicats du fichier BAM

Avec l'outil **MarkDuplicates** de la suite **PicardTools** intégrée à la suite **GATK4**

```
$ gatk MarkDuplicates --help          # affiche l'aide

$ sbatch -J markDup -o logs/markDup.out -e logs/markDup.err --mem=8G --wrap=" \
    gatk MarkDuplicates --java-options '-Xmx8G' \
    -I SRR1262731_extract.sort.rg.bam --VALIDATION_STRINGENCY SILENT \
    -O SRR1262731_extract.sort.rg.md.bam -M SRR1262731_extract_metrics_md.txt"
```

Marquage des duplicats de PCR

→ Garder les duplicats : 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…) ou Supprimer les duplicats du fichier BAM

```
$ sbatch -J flagstat2 -o logs/flagstat2.out -e logs/flagstat2.err --wrap=" \
    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      # % de pcrDup

# Bonus
$ grep -A1 "LIBRARY" SRR1262731_extract_metrics_md.txt | awk
'NR==2{printf("%.2f\n",$(NF-1)*100)}'
```

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
$ sbatch -J qualFilter -o logs/qualFilter.out -e logs/qualFilter.err --wrap=" \
    samtools view -bh -F 4 -q 30 SRR1262731_extract.sort.rg.md.bam \
    > SRR1262731_extract.sort.rg.md.filt.bam"

$ sbatch -J flagstat3 -o logs/flagstat3.out -e logs/flagstat3.err --wrap=" \
    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 mismatchs, de la taille d'insert, de paires mappées sur des chromosomes différents…

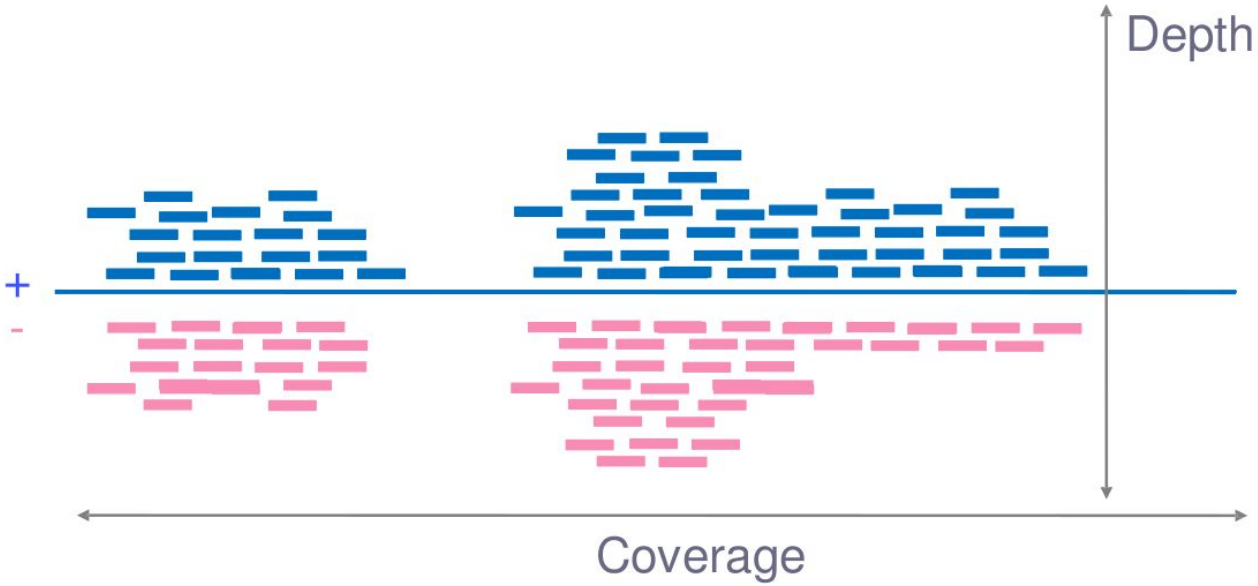
```
# Conservation des alignements dans les régions ciblées
$ module load bedtools/2.29.2
$ bedtools --version              # affiche la version (v2.29.2)
$ bedtools intersect --help       # affiche l'aide

$ sbatch -J interBed -o logs/interBed.out -e logs/interBed.err --wrap=" \
    bedtools intersect -a SRR1262731_extract.sort.rg.md.filt.bam \
    -b ~/tp_variant/additionnal_data/QTL_BT6.bed \
    > SRR1262731_extract.sort.rg.md.filt.onTarget.bam"

$ sbatch -J bamIndex -o logs/bamIndex.out -e logs/bamIndex.err --wrap=" \
    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

$ sbatch -J bamDepth -o logs/bamDepth.out -e logs/bamDepth.err --wrap=" \
    samtools depth -b ~/tp_variant/additionnal_data/QTL_BT6.bed \
    SRR1262731_extract.sort.rg.md.filt.onTarget.bam \
    > SRR1262731_extract.onTarget.depth.txt"

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