**GitHub repository title :**

A549\_cDNA\_Y2H\_library\_characterization\_workflow

**Git projetc repository (Git bash)**

francois.piumi@UC-1566 MINGW64 ~/OneDrive - ENVA/Bioinfo

$ cd **Projets/Banque\_A549/Workflow/**

francois.piumi@UC-1566 MINGW64 ~/OneDrive - ENVA/Bioinfo/Projets/Banque\_A549/Workflow

$ git config --global user.name "Francois Piumi"

francois.piumi@UC-1566 MINGW64 ~/OneDrive - ENVA/Bioinfo/Projets/Banque\_A549/Workflow

$ git config --global user.email [francois.piumi@inrae.fr](mailto:francois.piumi@inrae.fr)

francois.piumi@UC-1566 MINGW64 ~/OneDrive - ENVA/Bioinfo/Projets/Banque\_A549/Workflow

$ git config --global color.diff auto

francois.piumi@UC-1566 MINGW64 ~/OneDrive - ENVA/Bioinfo/Projets/Banque\_A549/Workflow

$ git config --global color.status auto

francois.piumi@UC-1566 MINGW64 ~/OneDrive - ENVA/Bioinfo/Projets/Banque\_A549/Workflow

$ git config --global color.branch auto

francois.piumi@UC-1566 MINGW64 ~/OneDrive - ENVA/Bioinfo/Projets/Banque\_A549/Workflow

$ git config --global core.editor vim

francois.piumi@UC-1566 MINGW64 ~/OneDrive - ENVA/Bioinfo/Projets/Banque\_A549/Workflow

$ git config --global merge.tool vimdiff

francois.piumi@UC-1566 MINGW64 ~/OneDrive - ENVA/Bioinfo/Projets/Banque\_A549/Workflow

$ git init

Initialized empty Git repository in C:/Users/francois.piumi/OneDrive - ENVA/Bioinfo/Projets/Banque\_A549/Workflow/.git/

Se placer dans ~/OneDrive - ENVA/Bioinfo/Projets/Banque\_A549/Workflow

git config --list

git status

git add 1.samtools\_index\_genome.sh

git status

git commit -m 1.samtools\_index\_genome.sh

git status

git remote add origin [git@github.com:fpiumi/A549\_cDNA\_Y2H\_library\_characterization\_workflow.git](mailto:git@github.com:fpiumi/A549_cDNA_Y2H_library_characterization_workflow.git)

git branch -M main

git push origin main

*Si erreur : To github.com:fpiumi/A549\_cDNA\_Y2H\_library\_characterization\_workflow.git*

*! [rejected] main -> main (fetch first)*

*error: failed to push some refs to 'github.com:fpiumi/A549\_cDNA\_Y2H\_library\_characterization\_workflow.git'*

*hint: Updates were rejected because the remote contains work that you do not*

*hint: have locally. This is usually caused by another repository pushing to*

*hint: the same ref. If you want to integrate the remote changes, use*

*hint: 'git pull' before pushing again.*

*hint: See the 'Note about fast-forwards' in 'git push --help' for details.*

Faire un

git pull --rebase origin main

et à nouveau un

git push origin main

**Input sequences:**

*merged.fastq.gz*

sequences.fastq.gz

**Step 1: genome indexation**

Script name: 1.samtools\_index\_genome.sh

**#!/bin/bash**

**module load bioinfo/samtools/1.19**

**samtools view faidx reference.fa**

output : reference.fa.fai

indexed genome is in the same folder as the reference genome

**Step 2: Read quality assement (Nanoplot)**

Script name: 2.nanoplot.sh

**#!/bin/bash**

**module load bioinfo/NanoPlot/1.42.0**

**NanoPlot --fastq sequences.fastq.gz**

**Step 3: genome mapping**

Script name: 3.minimap2\_genome.sh

**#!/bin/bash**

**module load bioinfo/Minimap/2-2.26**

**minimap2 -ax map-ont -N 100 \**

**reference.fa \**

**sequences.fastq.gz > minimap2\_genome.sam**

The –a argument specifies that the alignments should be output in SAM format (Sequence Alignment/Map), a standard format for representing sequence alignments.

The -x map-ont preset adjusts minimap2's internal parameters to handle the characteristics of ONT reads, which are long reads (tens to hundreds of thousands of bases long) and which may have a higher error rate compared to other sequencing technologies, such as Illumina. This preset ensures robust alignments despite these errors.

The -N parameter specifies the maximum number of secondary alignments reported per read. In genome alignment, the purpose is to map reads to the entire genomic sequence, which often contains repetitive elements, duplications, and homologous regions. Using -N 100 ensures to capture all possible mappings in a complex reference, especially in repetitive regions.

Command line:

sbatch --mem=60G minimap2\_genome.sh

output : minimap2\_genome.sam

**Step 4: splice-aware genome mapping**

Script name: 4.minimap2\_genome\_splice.sh

**#!/bin/bash**

**module load bioinfo/Minimap/2-2.26**

**minimap2 -ax splice –N 10 \**

**reference.fa \**

**sequences.fastq.gz > minimap2\_genome\_splice.sam**

For splice-aware alignments with cDNA reads, the goal is typically to align reads to their true genomic origin while considering exon-exon junctions. Using a -N 10 ensures to reduce secondary alignments to focus on relevant mappings and limit alignments to biologically meaningful regions.

**Step 5: transcriptome mapping**

Script name: 5.minimap2\_transcriptome.sh

**#!/bin/bash**

**module load bioinfo/Minimap/2-2.26**

**minimap2 -ax map-ont –N 100 \**

**reference.cdna.fa \**

**sequences.fastq.gz > minimap2\_transcriptome.sam**

**Step 6: transcriptome and the non-coding transcriptome concatenation mapping**

cdna+ncrna mapping :

Concatenation (cat command) of cdna and ncrna references-> **reference.cdna.ncrna.fa**

Remove duplicate sequences, otherwise an error is raised during the BAM compression

**awk '/^>/{f=!d[$1];d[$1]=1}f' reference.cdna.ncrna.fa > reference.cdna.ncrna.wo.duplicatesID.fa**

Script name: 6.minimap2\_coding\_noncoding\_transcriptome.sh

**minimap2 -ax map-ont -N 100 \**

**reference.cdna.ncrna.wo.duplicatesID.fa \**

**sequences.fastq.gz > minimap2\_transcriptome\_coding\_noncoding.sam**

**Step 7: sam to bam compression**

samtools\_sam2bam.sh

**#!/bin/bash**

**module load bioinfo/samtools/1.19**

**samtools view -bS minimap2\_genome.sam > minimap2\_genome.bam**

To be done for all the sam files

The -bS option in samtools view is used to specify that the input was in SAM format (-S) and that the output should be in BAM format (-b).

**Step 8: bam indexing/sorting**

Script 8.samtools\_sort\_index.sh

**#!/bin/bash**

**module load bioinfo/samtools/1.19**

**samtools sort minimap2\_genome.bam -o minimap2\_genome.sort.bam**

**samtools index minimap2\_genome.sort.bam**

output

**minimap2\_genome.sort.bam**

**minimap2\_genome\_splice.sort.bam**

**minimap2\_transcriptome.sort.bam**

**minimap2\_transcriptome\_coding\_noncoding.sort.bam**

**Step 9: samtools flagstat**

**#!/bin/bash**

**module load bioinfo/samtools/1.19**

**samtools flagstat minimap2\_genome\_sort.bam**

interprétation ?

**Step 10: Reads summarization**

Script 10.feature\_counts.sh

**#!/bin/bash**

**module load bioinfo/Subread/2.0.4**

**featureCounts -T 5 -t exon -g gene\_id \**

**-a reference.gtf \**

**-o minimap2\_genome.counts.txt \**

**minimap2\_genome.sort.bam**

also working with splice-aware genome mapping

**#!/bin/bash**

**module load bioinfo/Subread/2.0.4**

**featureCounts -T 5 -t exon -g gene\_id**

**-a reference.gtf \**

**-o minimap2\_genome\_splice.counts.txt \**

**minimap2\_genome\_splice.sort.bam**

Options :

-T specifies the number of threads to be used.

-t Specify the feature type. For example, features often correspond to exons and meta-features to genes

-g specify the attribute type used to group features (eg. exons) into meta-features. A meta-feature is the aggregation of a set of features. Here, meta-features are genes. The featureCounts program uses the gene\_id attribute available in the GTF format annotation (or the GeneID column in the SAF format annotation) to group features into meta-features, ie. features belonging to the same meta-feature have the same gene identifier.

-a is the genome annotation file (gtf file).

-o specifies the name of the output file, which includes the read counts

file.bam is an alignment file: in this file, the reads we want to count are aligned to the same genome as the annotation file.

Output :

counts.txt

For each meta-feature, the “Length” column gives the total length of genomic regions covered by features included in that meta-feature. the “Length” column typically contains the total number of non-overlapping bases in exons belonging to the same gene for each gene.

Strand : exons number of a gene

**Step 11: Reads summarization with transcriptome bam files**

feature counts is not working with transcriptome bam files, the following script must be used :

Script 11.samtools\_counts\_transcriptome\_files

**#!/bin/bash**

**module load bioinfo/samtools/1.19**

**samtools view minimap2\_transcriptome.sam \**

**-bh \**

**-t reference.cdna.fa \**

**-F 2324 > minimap2\_transcriptome.filt.bam**

**samtools sort minimap2\_transcriptome.filt.bam \**

**-o minimap2\_transcriptome.filt.sort.bam**

**samtools index minimap2\_transcriptome.filt.sort.bam**

**samtools view minimap2\_transcriptome.filt.sort.bam \**

**| cut -f 3 | sort | uniq -c**

**Step 12: Count files annotation**

An human genome annotation file is preliminarily downloaded from the biomart Ensembl Genes 112 database, (GRCh38.p14 human genes version) with the following attributes: Gene.stable.ID, Transcript.stable.ID, Gene.description, Gene.name, Gene.type

# annotation file opening

biomart <- read.csv2("mart\_export.txt",sep="\t")

library(dplyr)

biomart2 <- biomart %>% distinct(Gene.stable.ID, .keep\_all = TRUE)

# counts file opening

counts\_genome\_splice <- read.table( "minimap2\_genome\_splice.counts.txt" ,sep="\t", header = TRUE ,na.strings = c("Gene.name" , "Gene.type" ) )

counts\_genome\_splice\_short <- counts\_genome\_splice %>%

dplyr::select(-c("Chr" , "Start" , "End" , "Strand" , "Length"))

# variable name reduction: counts\_genome\_splice = GCS

GCS\_annotated <- merge(counts\_genome\_splice\_short, biomartbis, by.x="Geneid", by.y="Gene.stable.ID", all.x=TRUE)

# keep counts > 0

GCSA\_above\_zero <- GCS\_annotated[which(GCS\_annotated$minimap2\_genome\_splice.sort.bam > 0),]

# remove empty gene.names

CGSAAZ\_not\_empty <- GCSA\_above\_zero[!(GCSA\_above\_zero$Gene.name==""),]

# remove gene.names = NA

CGSAAZNE\_wo\_na <- na.exclude(CGSASupZ\_not\_empty)

# summarize counts

CGSAAZNE\_wo\_na\_grouped <- CGSASupZNV\_wo\_na %>%

group\_by(Gene.name,Gene.type) %>%

summarize(sum\_Counts = sum(minimap2\_genome\_splice\_sept24.sort.bam), #1

na.rm = TRUE) %>%

arrange(Gene.name)

# remove duplicated gene.names

CGSASupZNV\_wo\_na\_grouped2 <- CGSASupZNV\_wo\_na\_grouped %>%

group\_by(Gene.name) %>%

summarize(sum\_Counts = sum(sum\_Counts), #1

na.rm = TRUE) %>%

arrange(Gene.name)

## add again annotation because a part of it was lost during the previous steps

biomartbis3 <- biomart2[!(biomart2$Gene.name==""),]

biomartbis4 <- biomartbis3 %>%

group\_by(Gene.name) %>%

arrange(Gene.name)

CGSASupZNV\_wo\_na\_grouped3 <- merge(CGSASupZNV\_wo\_na\_grouped2,

biomartbis3 , by.x="Gene.name")

**Step 13: Functional Analysis**

Utilisation de ClusterProfiler sous R

# Gene list obtention to run Clusterprofiler

gene\_list <- CGSASupZNV\_wo\_na\_grouped3$Gene.name

# groupGO

library(ClusterProfiler)

ggo\_BP <- groupGO(gene = gene\_list,

OrgDb = org.Hs.eg.db,

keyType = 'SYMBOL',

ont = "BP",

level = 2,

readable = TRUE)

dim(ggo\_BP)

## GO terms

ego\_BP <- enrichGO(gene = gene\_list,

OrgDb = org.Hs.eg.db,

keyType = 'SYMBOL',

ont = "BP",

pAdjustMethod = "BH",

pvalueCutoff = 0.05,

qvalueCutoff = 0.05)

barplot(ego\_BP)

## GO terms reduction

library(GOSemSim)

library(enrichplot)

bp2 <- clusterProfiler::simplify(ego\_BP, cutoff=0.7, by="p.adjust", select\_fun=min)

d <- GOSemSim::godata('org.Hs.eg.db', ont="BP")

bp3 <- pairwise\_termsim(bp2, method = "Wang", semData = d)

treeplot(bp3)

# Reactome terms

# get entrez IDs

mart <- biomaRt::useEnsembl(biomart = "ensembl",

dataset = "hsapiens\_gene\_ensembl") #,

# mirror = "useast")

getbm\_res<-getBM(attributes = c("hgnc\_symbol","entrezgene\_id"), filters = "hgnc\_symbol",

values = CGSASupZNV\_wo\_na\_grouped3$Gene.name, mart = mart, uniqueRows = F)

getbm\_res$entrezgene\_id<-as.character(getbm\_res$entrezgene\_id)

CGSASupZNV\_wo\_na\_grouped3$entrezid <- getbm\_res$entrezgene[match(CGSASupZNV\_wo\_na\_grouped3$Gene.name,getbm\_res$hgnc\_symbol)]

library(ReactomePA)

packageVersion("ReactomePA")

reactome\_res <- enrichPathway(gene=CGSASupZNV\_wo\_na\_grouped2$entrezid, pvalueCutoff = 0.05, readable=TRUE)