Softwares and tools used in this study

Build new reference genome: *Tisochrysis lutea* v3

TrimGalore v0.6.2 performs filtration of short reads.

https://github.com/FelixKrueger/TrimGalore

```
trim_galore --suppress_warn --no_report_file --dont_gzip --max_n 0 --stringency 9 --length 150 --output_dir OutputDir --quality 30 --paired  RawRead_1 RawRead_2 --cores 8
```

InstaGrall, improved genome assembling with HIC-seq data.

briefly command used, see in github website for detail.

First step, hicstuff performs reference contact map from reference genome which will be improved.

https://github.com/koszullab/hicstuff

```
hicstuff pipeline -t CoresNumber -i -a bowtie2 -e DpnII -o Output-directory -g Reference-genome-v2.fasta HI-C_reads-1.fastq HI-C_reads-2.fastq
```

note : Used with 24 cores
note : -i option is for iterative alignment
note : DpnII is resctriction enzyme used

Second step, instagraal performs contact map, with 100 cycles.

https://https://github.com/koszullab/instaGRAAL

Warning: *require GPU to be used*

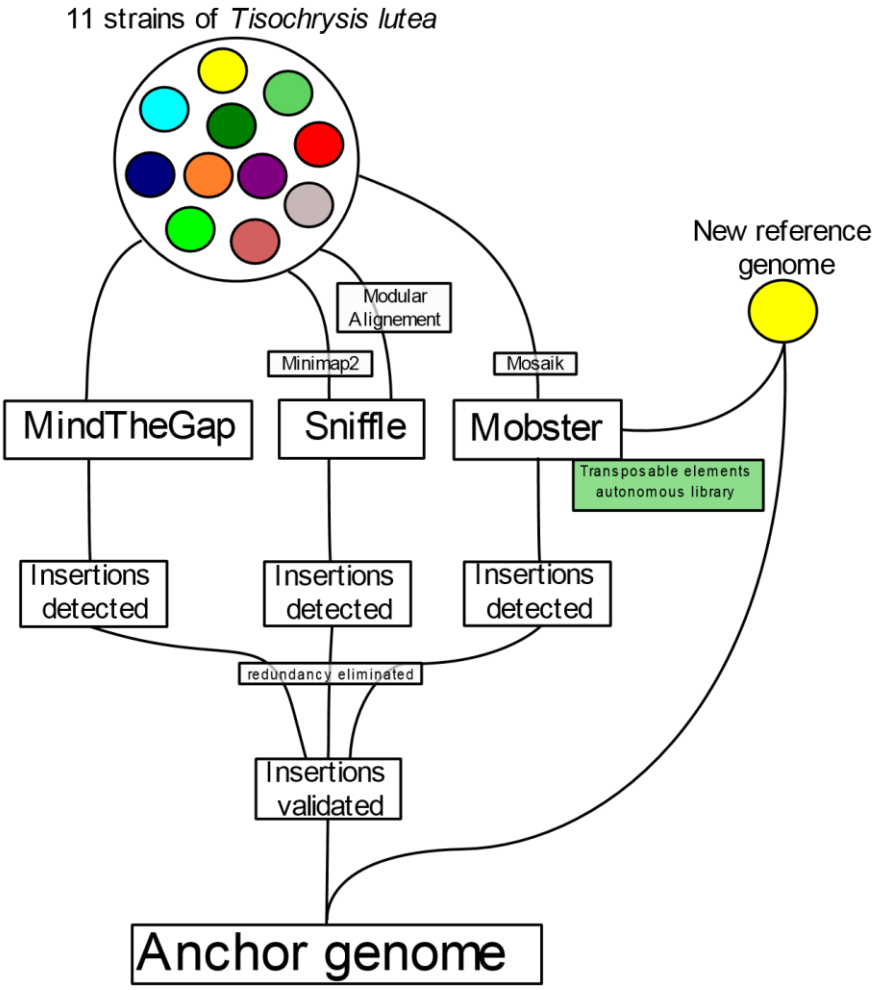
```
instagraal --save-matrix hicstuff_output-directory Reference-genome-v2.fasta  Output-directory
```

Third step, instagraal performs polishing of genome from results obtained previously.

```
instagraal-polish -m polishing -i info_frags.txt -f Reference-genome-v2.fasta -o Output_polishing
```

Note : info_frags.txt is the result of instalgrall with 100 cycles of contact maps.
Note :Output_polishing directory contained new reference-genome-v3.fasta

Anchor genome build



Alignment softwares used

BWA-MEM v0.7.17-r1188 perfoms alignment on genome from short read based in Burrows-Wheeler method.

```
bwa index Reference/anchor_genome.fasta

bwa mem  Reference/anchor_genome.fasta Filter-Read-1.fastq Filter-Read-2.fastq -R
"@RG\tID:$SampleName\tSM:$SampleName\tPL:ILLUMINA" -t CoresNumber >Output.sam

samtools view -b Output.sam >Output.bam
samtools sort Output.bam > Output-sort.bam
samtools index Output-sort.bam
```

note : SampleName is for exemple the name of strain strudy
note : Used with 24 cores
note : Samtools performs conversion sam in bam file, sort and index bam file

MiniMAP2 v2.17 performs alignment on genome from long read.

Index reference genome

```
minimap2 -x map-ont -d Reference_genome.mmi  Reference/anchor_genome.fasta
```

Alignment long reads on reference genome

```
minimap2 --MD -a -x map-ont -R "@RG\tID:$SampleName\tSM:$SampleName\tPL:MinION" -t
$CoresNumber Reference_Genome.mmi Long-reads-filtered.fq> Output.sam

samtools view -b Output.sam >Output.bam
samtools sort Output.bam > Output-sort.bam
samtools index Output-sort.bam
```

note : SampleName is for exemple the name of strain strudy
note : Used MinION option
note : Used with 24 cores
note : Samtools perform conversion sam in bam file, sort and index bam file

Mosaik v2.3 performs alignment on genome from short read based in Smith-Waterman method.

https://github.com/wanpinglee/MOSAIK

Index reference genome

```
MosaikBuild fr Reference_genome.fasta -sn Reference-genome-Sample -oa Output-ReferenceGenome.dat
```

Short reads index

```
MosaikBuild -q Short-Read1.fastq -q2 Short-Read2.fastq -st $SequencingMethod -mfl 400 -id SampleName -
sam SampleName -out Index-Reads.dat
```

Alignment

```
MosaikAligner -ia ReferenceGenome.dat -in Index-Reads.dat -out Output.bam -a multi -mhp 0 -p
CoresNumber -ls 200 -m all -zn -quiet -hs 15 -annpe 2.1.78.pe.ann -annse 2.1.78.se.ann

samtools sort Output.bam > Output-sort.bam
samtools index Output-sort.bam
```

note : SampleName is for exemple the name of strain strudy
note : Used with 24 cores
note : Samtools perform conversion sam in bam file, sort and index bam file

The Modular Aligner v1.1.1 performs alignment on genome from long read based in Smith-Waterman method.

Index reference genome

```
maCMD -X ReferenceGenome.fasta DirectoryReferenceGenome NameReferenceGenome
```

Alignement

```
maCMD p Nanopore -i  Long-reads-filtered.fq -x NameReferenceGenome.json -t $CoresNumber -o
Ouput.sam --Emulate_NGMLR's_tag_output true

samtools view -b Output.sam >Output.bam
samtools sort Output.bam > Output-sort.bam
samtools index Output-sort.bam
```

note : Used with 24 cores
note : Samtools perform conversion sam in bam file, sort and index bam file

Anchor genome, large insertions identification

Porechop v0.2.4 deletes adapters of long reads.
https://github.com/rrwick/Porechop

```
porechop -i Raw-long-reads.fastq -o Output_reads.fastq --adaptater_threshold 95 -–middle_threshold 90 –-end_threshold 85
```

Nanofilt v2.6.0 performs filtration of long reads.
https://github.com/wdecoster/nanofilty

```
NanoFilt Raw-long-reads.fastq –quality 8 –length 2000 > Filtered-long-reads.fastq
```

Sniffles v1.0.11 performs large insertions detection from long read sequenced.
https://github.com/fritzsedlazeck/Sniffles

```
sniffles -m Alignement-File.bam -v Output.vcf --min_support 10  --threads CoresNumber --min_length 1000 --minmapping_qual 25 --report_seq
```

note : min support option is number of minimal reads to consider a large insertion
note : Alignement-File.bam provide of long reads map with Minimap2 or Modular Aligner
note : Used with 8 cores

Mind The Gap v2.2 performs detection and assembly larges insertion variants from directly short reads sequenced.
https://github.com/GATB/MindTheGap

```
MindTheGap find -in Short-Read-1.fastq, Short-Read-2.fastq -ref Reference-Genome-v3.fasta -out Output-Directory  -nb-cores CoresNumber -max-memory RamMemory
```

```
MindTheGap fill -graph Output-Graph.h5 -bkpt Output.breakpoints -out Output.vcf -nb-cores CoresNumber -max-memory RamMemory  -max-length 20000 -max-nodes 300
```

note : Used with 8 cores
note : Used with 10 gigaoctet in RAM

Mobster v0.2.4.1 performs detection of transposable elements known.
https://github.com/jyhehir/mobster

```
java -Xmx48G -jar -jar MobileInsertions-0.2.4.jar -properties OptionsMobster.conf -in Alignement-File.bam -out Output-MobsterPrediction.txt -sn SampleName
```

```
java -jar MobsterVCF.jar -file Output-MobsterPrediction.txt -out $ Output-Mobster.vcf
```

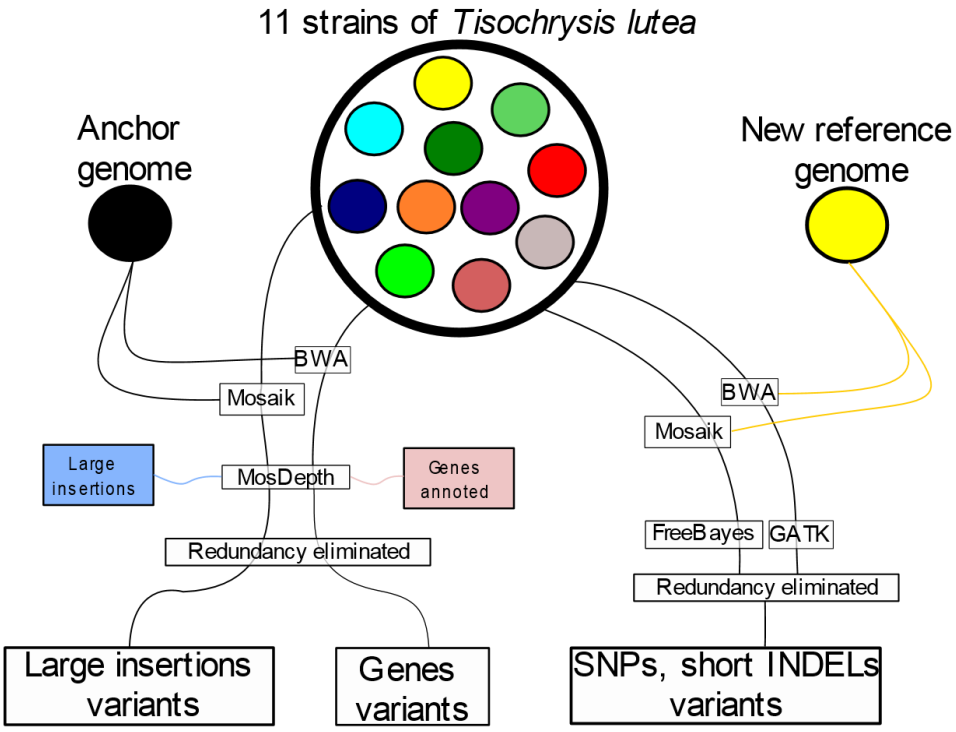
note : The configuration file contained link from transposable elements in fasta format ; Read length is fixed at 150 bases and minimum supporting read is fixed at 10.

VCFlib v1.0.2 performs consensus genome from VCF file and genome in fasta.
https://github.com/vcflib/vcflib

```
vcf2fasta -f Tiso_Reference_Genome_V3.0.fasta -P InsertionVariants.vcf > Tiso_Anchor_Genome.fasta
```

Note : InsertionVariant.vcf is split by contigs

Polymorphism identification into pan-genomes



Freebayes v1.3.1 performs detection of small polymorphisms SNPs (single-nucleotide polymorphisms), indels (insertions and deletions).

https://github.com/freebayes/freebayes

```
freebayes -L All-Aligmenent-Files.bam -f Reference-Genome-v3.fasta -v OutputSortie.vcf --min-mapping-quality 20 --min-base-quality 30 --min-alternate-fraction 0.10 --min-alternate-count 5 --min-coverage 40
```

note : used in same time with all alignment files from each strains

GATK v4.1.3.0 performs detection of small polymorphisms.

https://gatk.broadinstitute.org/hc/en-us

Index reference genome

```
java -jar gatk-package-4.1.3.0-local.jar CreateSequenceDictionary --REFERENCE Reference-Genome-v3.fasta
```

Calling polymorphism

```
java -jar gatk-package-4.1.3.0-local.jar HaplotypeCaller –output Output.vcf --minimum-mapping-quality 20 --min-base-quality-score 30 --reference Reference-Genome-v3.fasta --max-alternate-alleles 2 --input All-Aligmenent-Files.bam
```

note : used in same time with all alignment files from each strains

Mosdepth v0.2.9 performs depth calculation from bam file.

https://github.com/bentp/mosdepth

```
mosdepth Output-PREFIX File.bam --by BedFile --thresholds 1 --threads 8
```

note :BedFile is annotation of transposable elements or genes annotation in anchor genome