Formation sRNAseq - Analyse des miRNA

- EXERCICES -

"**FastQC** is a quality control tool for high throughput sequence data."
http://www.bioinformatics.bbsrc.ac.uk/

cutadapt

A tool that removes adapter sequences from DNA sequencing reads. ***cutadapt*** removes adapter sequences from high-throughput sequencing data. https://code.google.com/p/cutadapt/

B W A

"**Burrows-Wheeler Aligner** (**BWA**) is an efficient program that aligns relatively short nucleotide sequences against a long reference sequence such as the human genome." http://bio-bwa.sourceforge.net

SAMtools

"**SAM** (Sequence Alignment/Map) format is a generic format for storing large nucleotide sequence alignments."
http://samtools.sourceforge.net

Objectifs :

Cette formation a pour objectif de vous aider à traiter les séquences issues de projet de sRNAseq (miRNA). Vous y découvrirez les problématiques spécifiques de l'analyse des petits ARNs non codant, les outils liés et les mettrez en œuvre afin de détecter, annoter, prédire, quantifier, … les miRNA.

Pré-requis : savoir utiliser un environnement Unix et avoir suivi la formation « alignement de séquence issues des SGS et recherche de polymorphismes ».

Pour réaliser l'ensemble de ces exercices connecter vous sur votre compte « genotoul » (en utilisant « putty » et « xming » sous windows).

Pour les traitements « lourds » utiliser le cluster.

Sur « genotoul », créer dans le répertoire work un répertoire de travail F13f :

```
mkdir ~/work/F13f ; cd F13f
```

Exercice n°1 : Analyse de la qualité – suppression des adaptateurs – suppression de la redondance intra et inter « fastq »

Quelques liens :

- FastQC : http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc
- Cutadapt

« cutadapt » est déjà installés sur « genotoul ».
Penser à utiliser la complétion en utilisant la touche 'Tab'. (exemple : cuta + 'Tab')

Première étape : créer et explorer l'environnement de travail

```
cp -r /save/sigenae/F13f/TP ~/work/F13f/
cd TP
ls *
```

Nous allons travailler à partir de 5 « lanes » Illumina représentant 5 tissus :

- Copier, dans le répertoire « data », 5 fichiers « fastq » disponibles dans le répertoire /save/sigenae/F13f/Fastq :
  ```
  cp /save/sigenae/F13f/Fastq/*fastq data/
  ```
- Appréhender le résultat du séquençage à l'aide de « FastQC » :
  - Produire un fichier de commandes pour les 5 « fastq »
    ```
    foreach i ( data/*.fastq )
    echo "bin/FastQC/fastqc -o 1_fastqc $i" >> 1_fastqc.jobs
    end
    ```
  - Lancer l'exécution sur le cluster :
    ```
    qarray -o 1_fastqc -e 1_fastqc -N fastqc 1_fastqc.jobs
    ```
  - Visualiser les rapports « html ».

Suppression des adaptateurs à l'aide de « cutadapt » (adaptateur utilisé : `ATCTCGTATGCCGTCTTCTGCTTG` – taille minimum 16pb – taille maximum 28pb) :

- Afficher l'aide de la commande « cutadapt » :
  ```
  cutadapt -h
  ```
- Produire et exécuter sur le cluster le fichier de commandes :
  ```
  cd data
  foreach i ( *fastq )
  echo "(cutadapt -a ATCTCGTATGCCGTCTTCTGCTTG -m 16 -M 28 -o 2_cutadapt/$i.cut.fq data/$i) >& 2_cutadapt/$i.cut.log" >> ../2_cutadapt.jobs
  end
  cd ..
  qarray -o 2_cutadapt -e 2_cutadapt -N cadapt 2_cutadapt.jobs
  ```

Statistiques « cutadapt » (nombre de séquence – distribution des longueurs de lecture) :

- Utiliser le script « stat1.pl » sur le cluster, afin de produire le fichier de statistiques (« .txt »)

```
bin/stat1.pl
```

```
qsub -b y -o stat/ -e stat/ -N stat 'bin/stat1.pl 2_cutadapt/*fq > 2_cutadapt/2_stat.txt'
```

- Utiliser le script « plot1.pl » afin de produire le rapport « html »

```
bin/plot1.pl
```

```
bin/plot1.pl 2_cutadapt/2_stat.txt > stat/2_stat.html
```

Élimination de la redondance pour chaque tissus :

- Utiliser le script « fastqnr.pl », produire et exécuter sur le cluster le fichier de commandes (penser à trier la sortie du script sur le premier champ) :

```
cd 2_cutadapt
foreach i (*fq)
  echo "bin/fastqnr.pl 2_cutadapt/$i | sort -k1,1 > 3_redundancy/$i.nr" >> ../3_redundancy.jobs
end
cd ..
```

```
qarray -N redundancy -o 3_redundancy/ -e 3_redundancy/ 3_redundancy.jobs
```

- Afficher les premières lignes des fichiers de sortie « .nr »

Construction d'une matrice à partir de l'ensemble des séquences des différents « fastq » et filtre :

- Utiliser le script « join.pl » sur le cluster :

```
bin/join.pl
```

```
qsub -b y -o 3_redundancy/ -e 3_redundancy/ -N join 'bin/join.pl 3_redundancy/*.nr > 3_redundancy/ALL.matrix'
```

- Utiliser le script « matrix.pl » sur le cluster, afin de filtrer la matrice et produire trois fichiers : « .fasta », « .txt » et un fichier contenant la matrice d'expression filtrée « .csv ».

  Paramétrer le script pour ne conserver que les séquences présentent en 10 copies minimum (pour un « fastq ») **ou** présentent dans 3 « fastq » minimum :

```
bin/matrix.pl
```

```
qsub -b y -o 3_redundancy/ -e 3_redundancy/ -N matrix 'bin/matrix.pl -a 10 -i 3 3_r*/ALL.matrix -o 3_redundancy/ALL.matrix'
```

- Explorer les fichiers générés (comment sont formatés les identifiants de séquence?) :

```
head 3_redundancy/ALL.matrix.*
```

- Utiliser le script « plot2.pl » afin de produire le rapport « html »

```
bin/plot2.pl 3_redundancy/ALL.mat*.txt > stat/3_stat.html
```

Exercice n°2 : Annotations

Quelques liens (outils / base de données) :

- BWA : http://bio-bwa.sourceforge.net
- BWA man : http://bio-bwa.sourceforge.net/bwa.shtml
- SAMtools : http://samtools.sourceforge.net
- mdust : http://compbio.dfci.harvard.edu/tgi/software/

- mirBase => ftp://mirbase.org/pub/mirbase/CURRENT/ (hairpin.fa.gz)
- Rfam => ftp://ftp.sanger.ac.uk/pub/databases/Rfam/CURRENT/ (Rfam.fasta.gz)
- tRNA => http://gtrnadb.ucsc.edu/download.html (eukaryotic-tRNAs.fa.gz)
- rRNA => ftp://ftp.arb-silva.de/current/Exports/ ([LS]SURef_108_tax_silva_trunc.fasta.tgz)

Alignement des séquences contre chaque base de données avec « BWA » :

- Créer, dans le répertoire « bank », les liens symboliques vers l'ensemble des fichiers disponibles dans le répertoire « /save/sigenae/data/miRNA/bank/ » :

```
ln -s /save/sigenae/F13f/Bank/* bank
```

Les différentes banques ont été préalablement indexées par « BWA » :

```
bwa index -a bwtsw bank.fasta
```

- Produire et exécuter sur le cluster le fichier de commandes (alignement, « bam », tri) :

```
cd bank
foreach i ( *fa *fasta )
 echo "bwa aln bank/$i 3_redundancy/ALL.matrix.filter.tfa | bwa samse bank/$i - 3_redundancy/ALL.matrix.filter.tfa | samtools view -bS - | samtools sort - 4_annotation/$i.sort" >> ../4_annotation.jobs
end
cd ..
  qarray -N bwa -o 4_annotation/ -e 4_annotation/ 4_annotation.jobs
```

- Filtrer les alignements (Flag 0 ou 16 – 1 mismatch autorisé) et produire deux fichiers (le premier contenant les colonnes « qname » et « rname » trier sur « qname » et le second identique mais ne contenant que la colonne « qname ») :

```
foreach i (4_annotation/*.sort.bam)
 samtools view $i | grep 'NM:i:[01]' | awk '$2==0||$2==16' | cut -f1,3 | sort -k1,1 > $i.filter1
 cut -f1 $i.filter1 > $i.filter2
end
```

Analyse de l'annotation :

- Produire une matrice d'annotation à l'aide du script « join.pl » :

```
bin/join.pl 4_annot*/*filter1 > 4_annotation/annot.csv
```

- Comparaison des annotations – diagrammes de Venn, à l'aide du script « lists2venn.pl » :

```
bin/lists2venn.pl 4_ann*/euk*filter2 4_ann*/LSU*filter2 4_ann*/SSU*filter2 stat/4_tRNA-rRNA
```

```
cat 4_an*/euk*filter2 4_ann*/LSU*filter2 4_ann*/SSU*filter2 | sort -u > 4_annotation/tRNA-rRNA.filter2
```

```
bin/lists2venn.pl 4_ann*/tRNA-rRNA*2 4_ann*/Rfam*filter2 4_ann*/hairpin*filter2 stat/4_rtRNA-mir-Rfam
```

Les images générées peuvent être visualisées directement depuis la machine « genotoul » avec la commande : `display img.png`

- Joindre la matrice d'annotation avec la matrice d'expression filtrée (attention les deux fichiers doivent être trié sur le champ permettant la jointure) :

```
sort -k1,1 3*/*csv > 3_redundancy/ALL.matrix.filter.csv.sort
```

```
join -t 'ctrl-v <tab>' -1 1 -2 1 4*/annot.csv 3*/*.csv.sort > 4_annotation/all.csv
```

Pour insérer une « vraie tabulation » utiliser la combinaison de touches : `ctrl-v <tab>`

- Quelles sont les 10 séquences les plus exprimées :

```
sort -k8,8nr 4_annotation/all.csv | head
```

- Combien de séquences sont spécifiques d'un seul « fastq » :

```
grep -c '#1#' 4_annotation/all.csv
```

- Pour cet ensemble (spécifiques d'un « fastq »), combien sont annotées dans mirBase :

```
perl -lane 'print $_ if($F[2] ne "0" && $F[0]=~/#1#/)' 4_annotation/all.csv | wc -l
```

- Filtrer la matrice afin d'obtenir les séquences annotées par mirBase et Rfam, et présentent en 100 copies minium :

```
perl -lane 'print $_ if($_ =~ /^#/ || ($F[2] ne "0" && $F[4] ne "0" && $F[7]>100))' 4_annotation/all.csv > 4_annotation/all_filter.csv
```

- [...]

- Utiliser le script « matrix2html.pl » afin de produire le rapport « html » :

```
bin/matrix2html.pl
```

```
bin/matrix2html.pl 4_an*/all_filter.csv > stat/4_stat.html
```

Attention : le rapport « html » est limité aux 10 000 premières lignes du fichier fourni.

Exercice n°3 : Alignement sur référence

Alignement des séquences contre la référence avec « BWA » :

- Créer, dans le répertoire « 5_reference », les liens symboliques vers l'ensemble des fichiers disponibles dans le répertoire « /save/sigenae/F13f/Reference/ » :

```
ln -s ~/save/F13f/Reference/* 5_reference/
```

La référence a été préalablement indexées par « BWA » :

```
bwa index -a bwtsw bank.fasta
```

- Exécuter sur le cluster l'alignement : :

```
qsub -b y -o 5_reference/ -e 5_reference/ -N bwa 'bwa aln 5_ref*/V4_454Scaffolds.fna 3_red*/ALL.matrix.filter.tfa | bwa samse -n 100 5_ref*/V4_454Scaffolds.fna - 3_red*/ALL.matrix.filter.tfa | samtools view -bS - | samtools sort - 5_reference/ALL.matrix.filter.sort'
```

- Statistiques d'alignement – utiliser les scripts « bwasam2stat.pl » et « plot3.pl » afin de produire le rapport « html » :

```
samtools view 5_reference/ALL.matrix.filter.sort.bam | bin/bwasam2stat.pl - > 5_reference/5_stat1.txt
bin/plot3.pl 5_reference/5_stat1.txt > stat/5_stat1.html
```

Un autre rapport « html » d'analyse de l'alignement « bam » peut être obtenu grâce à « samstat » installé sur « genotoul »

```
samstat 5*/ALL.matrix.filter.sort.bam -n stat/5_stat
```

- Filtrer l'alignement enfin de ne conserver uniquement les lectures alignées sans INDEL, avec un maximum de 1 erreur (NM:i:[01]), 3 localisations « best hits » (X0:i:[123]) et 95 localisations « suboptimal hits » (X1:i:<=95) – « bwasamXfilter.pl » :

```
samtools view -h 5_reference/ALL.matrix.filter.sort.bam | ( bin/bwasamXfilter.pl - -n 1 -0 3 -1 95 -I -o > 5_reference/ALL.matrix.filter.NMX.sam ) >& 5_reference/5_stat2.txt
```

- Statistiques de filtrage – utiliser le script « plot4.pl » afin de produire le rapport « html » :

```
bin/plot4.pl 5_reference/5_stat2.txt > stat/5_stat2.html
```

- Statistiques d'alignement – utiliser les scripts « bwasam2stat.pl » et « plot3.pl » afin de produire le rapport « html » :

```
bin/bwasam2stat.pl 5*/A*NMX.sam > 5_reference/5_stat3.txt
bin/plot3.pl 5_reference/5_stat3.txt > stat/5_stat3.html
```

- A partir de l'alignement filtré, utilisation du tag « XA » afin de dupliquer les lignes pour les lectures multi-localisées :

```
bin/samNoXA.pl 5*/ALL.matrix.*.NMX.sam | samtools view -bS - | samtools sort - 5_reference/ALL.matrix.filter.NMX.NOXA
```

Exercice n°4 : Identification des locus

A partir du BAM :

- Utiliser le script « sam2locus.pl » afin d'identifier les locus et les potentiels miRNA-miRNAstar

```
samtools view 5_reference/ALL.matrix.filter.NMX.NOXA.bam | ( bin/sam2locus.pl 5_reference/V4_454Scaffolds.fna > locus_miRNA-miRNAstar ) > & locus
```

- Explorer les fichiers générés
- Extraire les locus pour lesquels le miRNA et le miRNAstar sont exprimés

```
perl -lne 'BEGIN{$/="\n##\n"} print "##\n$_" if ($_ =~ /locusNumber:2/)' locus_miRNA-miRNAstar > locus_miRNA-miRNAstar_2
```