Tutorial: Whole Genome Bisulfite Sequencing Pipeline

Introduction

This tutorial was created by Dr. Mehran Karimzadeh for MMG3003Y, and adapted by Dr. Martina Steiner. You will work through a series of quizzes on Quercus, and go through a pipeline of software on the teach cluster. This will prepare you for this week's assignment.

Work through the following Quizzes now: 1. EpigeneticsII_WGBS-context 2. EpigeneticsII_methods 3. EpigeneticsII_bisulfite-conversion 4. EpigeneticsII_computational-steps

Background 1: bash

- Log into the teach cluster as described in the RNAseq tutorial
- Avoid computation on the login node #describe why
- Request an interactive session

```
salloc
samtools ...
```

- Write a bash script and submit using *sbatch*

```
echo -e '#!/bin/sh' > myBashScriptThatIWillSubmitToRunOnSlurm.sh
echo "samtools ..." >> myBashScriptThatIWillSubmitToRunOnSlurm.sh
sbatch -c 1 --mem=4G -t 1:00:00 myBashScriptThatIWillSubmitToRunOnSlurm.sh
```

- Work through the following Quiz on Quercus now

4. EpigeneticsII_bash
5. EpigeneticsII_pipeline

Background 2: Bisulfite sequencing pipeline

- Pipeline outline
  - Download the reference genome
  - Index the reference genome
  - Copy fastq files from a shared folder
  - Trimming fastq files with trim galore
  - Aligning with bwa-meth (copy the BAM files from a shared folder)
  - Extracting CpG methylation estimates
  - Visualizing methylation at CpG islands (supplementary)

  - Identifying differentially methylated regions
  - Investigating genes affected by differential methylation

- Tutorial steps:

  - Tutorial step 1: Data access, trimming and quality control
  - Tutorial step 2: Alignment
  - Tutorial step 3: Extracting estimates
  - Tutorial step 4: Identify differentially methylated regions

Tutorial step 1: Data access - Indexing BWA-meth

- Download FASTA file of chr22

```
mkdir -p $SCRATCH/Ref
cd $SCRATCH/Ref
wget ftp://ftp.ensembl.org/pub/release-96/fasta/homo_sapiens/dna/Homo_sapiens.GRCh38.dna.chromosome.22.
zcat Homo_sapiens.GRCh38.dna.chromosome.22.fa.gz | sed 's/>22/>chr22/' > Homo_sapiens.GRCh38.dna.chromos
```

- Index FASTA file

```
salloc
module load anaconda3 gcc java fastqc cutadapt trimgalore bwa samtools bwameth htslib methyldackel
bwameth.py index Homo_sapiens.GRCh38.dna.chromosome.22.fa
```

- Expected run time is 3 minutes

Tutorial step 1: Data access - Download CpG islands

- Visit http://genome.ucsc.edu
- Select the 3rd tool, *Table Browser*
- Select the correct genome assembly, and group *Regulation*
- Select the *CpG Islands* track
- Under position, type chr22
- Select the output format as *BED - browser extensible data*
- Save the file as: *hg38CpgIslandsForChr22.bed*
- Use scp to transfer the file to $SCRATCH/Datasets

```
# From the local computer
scp ~/Downloads/hg38CpgIslandsForChr22.bed username@teach.scinet.utoronto.ca:/scratch/t/teachmmg3003/mm
```

Tutorial step 1: Data access - Downloading necessary files

- The following directory contains all of the pipeline for chromosome 22:

```
/scratch/m/mhoffman/karimzad
```

- You can copy the fastq files from that directory to your scratch folder:

```
cp -rf /scratch/m/mhoffman/karimzad/newFastqFilesChr22 $SCRATCH
```

The reads in this file are only from chromosome 22 to cut down on time required for computation. For the assignment, you will repeat the sequence of steps for chromosome 21.

- This folder also contains all of the files from the pipeline we process. You can copy them the same way if you have issues running the commands.

What are the samples?

- H1-hESC is a human embryonic stem cell line which has been profiled extensively by the ENCODE consortium
- The left ventricle embryonic tissue is obtained from a human embryo
- By comparing these two tissues, we *may* identify which regions of chr22 must be (de)methylated for differentiating the embryonic stem cell towards a heart muscle progeny.

##Tutorial step 1: Trimming - Trim the FASTQ files

- Write a script to trim paired-end FASTQ files with trim galore in a new folder called trimmedFastqs

```
LOGDIR=$SCRATCH/Logs # Creates a new variable
SCRIPTDIR=$SCRATCH/Scripts # For scripts that run different programs
mkdir -p $SCRIPTDIR $LOGDIR # Creating multiple folders simultaneously
cd $SCRIPTDIR # Change diretory
FASTQDIR=$SCRATCH/newFastqFilesChr22 # PAth to our untrimmed fastq files
FQFOLDERS=($(ls $FASTQDIR)) # Arrays
OUTMAIN=$SCRATCH/trimmedFastqsChr22
for FQFOLDER in ${FQFOLDERS[@]}
do
  FQ1=$FASTQDIR/$FQFOLDER/$FQFOLDER\__1.fastq.gz
  FQ2=$FASTQDIR/$FQFOLDER/$FQFOLDER\__2.fastq.gz
  OUTDIR=$OUTMAIN/$FQFOLDER
  mkdir -p $OUTDIR
  echo -e '#!/bin/sh' > $SCRATCH/Scripts/$FQFOLDER\_TrimGalore.sh
  echo "module load anaconda3 gcc java fastqc cutadapt trimgalore bwa samtools bwameth htslib methyldac
  echo "trim_galore --fastqc --paired --gzip -o $OUTDIR $FQ1 $FQ2" >> $SCRATCH/Scripts/$FQFOLDER\_TrimG
  sbatch -c 1 -t 1:00:00 -e $LOGDIR/$FQFOLDER\_TrimGalore.%A.ERR -o  $LOGDIR/$FQFOLDER\_TrimGalore.%A.L
done
```

Tutorial step 1: Quality control - FASTQC reports

- Use *scp* to download fastqc files to your local computer

```
STUDENTID=05
scp -r mmg3003student$STUDENTID@teach.scinet.utoronto.ca:/scratch/t/teachmmg3003/mmg3003student$STUDENT
```

Tutorial step 2: Alignment - Align with BWA-Meth

- Write a script to generate _Align.sh scripts for aligning fastq files and submit them to cluster with sbatch

```
cd $SCRIPTDIR
REF=$SCRATCH/Ref/Homo_sapiens.GRCh38.dna.chromosome.22.fa
FASTQDIR=$SCRATCH/trimmedFastqsChr22
BAMDIR=$SCRATCH/trimmedAlignedBamsChr22
mkdir -p $BAMDIR
mkdir -p $SCRIPTDIR
SAMPLES=($(ls $FASTQDIR))
for SAMPLE in ${SAMPLES[@]}
do
  FQ1=$(ls $FASTQDIR/$SAMPLE | grep val_1.fq.gz)
  FQ2=$(ls $FASTQDIR/$SAMPLE | grep val_2.fq.gz)
  echo -e '#!/bin/sh' > $SCRIPTDIR/$SAMPLE\_Align.sh
  echo "module load anaconda3 gcc java fastqc cutadapt trimgalore bwa samtools bwameth htslib methyldac
  echo "bwameth.py --reference $REF $FASTQDIR/$SAMPLE/$FQ1 $FASTQDIR/$SAMPLE/$FQ2 | samtools view -bS -
  # sbatch -c 1 -t 4:00:00 -e $LOGDIR/$SAMPLE\_Align.%A.ERR -o $LOGDIR/$SAMPLE\_Align.%A.LOG $SCRIPTDIR
done
```

Tutorial step 2: Alignment - Copy the aligned bam files

- It takes 4 hours of CPU time to align the FASTQ files to chr22.
- Assignment: Similar to FASTQ files, copy the folder containing bam files of chr22 to your *$SCRATCH* NOW!

Tutorial step 2: Alignment - Sort and index bam files

- MethylDackel requires sorted and indexed bam files
- Write a script to sort and index each bam file

```
cd $SCRIPTDIR
BAMDIR=$SCRATCH/trimmedAlignedBamsChr22
BAMFILES=($(ls $BAMDIR | grep .bam | grep -v bam.bai | grep -v sorted))
for BAMFILE in ${BAMFILES[@]}
do
  SAMPLENAME=$(echo $BAMFILE | sed 's/.bam//')
  echo -e '#!/bin/sh' > $SCRATCH/Scripts/$SAMPLENAME\_sortAndIndex.sh
  echo "module load anaconda3 gcc java fastqc cutadapt trimgalore bwa samtools bwameth htslib methyldac
  echo "samtools sort $BAMDIR/$BAMFILE -o $BAMDIR/$SAMPLENAME\_sorted.bam" >> $SCRATCH/Scripts/$SAMPLEN
  echo "samtools index $BAMDIR/$SAMPLENAME\_sorted.bam" >> $SCRATCH/Scripts/$SAMPLENAME\_sortAndIndex.s
  sbatch -c 1 -t 1:00:00 -e $LOGDIR/sortIndex.%A.ERR -o $LOGDIR/sortIndex.%A.LOG $SCRATCH/Scripts/$SAMP
done
```

Tutorial step 3: Extracting estimates - Run MethylDackel

- MethylDackel uses BAM files to extract cytosine methylation counts
- Run a script to run MethylDackel files for each BAM file

```
cd $SCRIPTDIR
BAMDIR=$SCRATCH/trimmedAlignedBamsChr22
OUTMAIN=$SCRATCH/methylDackelOutputChr22
BAMFILES=($(ls $BAMDIR | grep sorted | grep -v bai | grep bam))
REF=$SCRATCH/Ref/Homo_sapiens.GRCh38.dna.chromosome.22.fa
for BAMFILE in ${BAMFILES[@]}
do
  SAMPLENAME=$(echo $BAMFILE | sed 's/_sorted.bam//')
  OUTDIR=$OUTMAIN/$SAMPLENAME
  mkdir -p $OUTDIR
  echo -e '#!/bin/sh' > $SCRIPTDIR/MethylDackel_$SAMPLENAME.sh
  echo "module load anaconda3 gcc java fastqc cutadapt trimgalore bwa samtools bwameth htslib methyldack
  echo "MethylDackel extract --fraction --mergeContext $REF $BAMDIR/$BAMFILE -o $OUTDIR/$SAMPLENAME\_" >
  sbatch -c 1 -t 1:00:00 -e $LOGDIR/Meth.%A.ERR -o $LOGDIR/Meth.%A.LOG $SCRIPTDIR/MethylDackel_$SAMPLENA
done
```

Tutorial step 3: Extracting estimates - Explore the output of MethylDackel

- What does each column of MethylDackel output represent?

```
track type="bedGraph" description="/scratch/t/teachmmg3003/mmg3003ta002/methylDackelOutput/H1-hESC_rep1/
22      10513864        10513865        0
22      10513906        10513907        1
22      10515169        10515170        0
```

Tutorial step 3: Extracting estimates - bedGraph is not efficient

- bedGraph is a user-readable file format
- Storing genomic signal in bedGraph format takes too much space and is computationally inefficient for random data retrieval
- bigWig format, however, can store and retrieve genomic signals efficiently
- Here we will download a program called bedGraphToBigWig and use it to convert bedGraph files

```
mkdir -p ~/software/bin
cd ~/software/bin
wget http://hgdownload.soe.ucsc.edu/admin/exe/linux.x86_64/bedGraphToBigWig
# Give yourself permission to run this program
chmod u+x bedGraphToBigWig
```

Tutorial step 3: Extracting estimates - Finding size of chromosomes

- bedGraphToBigWig requires a file with information of how long each chromosome is

```
cd ~/software/bin
wget http://hgdownload.soe.ucsc.edu/admin/exe/linux.x86_64/fetchChromSizes
chmod u+x fetchChromSizes
./fetchChromSizes hg38 > $SCRATCH/Ref/hg38.chromsizes
```

- How else can we extract chromosome sizes from a fasta file?

Tutorial step 3: Extracting estimates - Convert bedGraph to bigWig

- Write a script to convert output of MethylDackel from bedGraph to bigWig

```
salloc
MAINDIR=$SCRATCH/methylDackelOutputChr22
SAMPLES=($(ls $MAINDIR))
for SAMPLE in ${SAMPLES[@]}
do
  BDG=$(ls $MAINDIR/$SAMPLE | grep bedGraph)
  BW=$(echo $BDG | sed 's/bedGraph/bigWig/')
  ~/software/bin/bedGraphToBigWig $MAINDIR/$SAMPLE/$BDG $SCRATCH/Ref/hg38.chromsizes $MAINDIR/$SAMPLE/$
done
```

Tutorial step 4: Identify DMRs - Identify differentially methylated regions

- There are various software for identifying differentially methylated regions
- Here we will use https://dx.doi.org/10.1101%2Fgr.196394.115
- Metilene requires a union file of bedGraphs we generated earlier with MethylDackel with the following columns:

```
Chrom  Start  End  G1_1  G1_2  G2_1  G2_2
```

- We can generate the input file this way:

```
MAINDIR=$SCRATCH/methylDackelOutputChr22
SAMPLES=($(ls $MAINDIR))
BGS=()
HEADER=(chr start end)
for SAMPLE in ${SAMPLES[@]}
do
  HEADER+=($SAMPLE)
  BG=$(ls $MAINDIR/$SAMPLE | grep bedGraph)
  BGS+=($MAINDIR/$SAMPLE/$BG)
done
module load gcc/7.3.0 bedtools
echo -e ${HEADER[@]} | tr " " "\t" > $SCRATCH/methylDackelOutputChr22/mergedOutputs_unionbedg.bed
bedtools unionbedg -i ${BGS[@]} >> $SCRATCH/methylDackelOutputChr22/mergedOutputs_unionbedg.bed
```

Tutorial step 4: Identify DMRs - Metilene

```
module load metilene
OUTDIR=$SCRATCH/metileneOutputChr22
mkdir -p $OUTDIR
echo -e "Chrom\tStart\tEnd\tqVal\tmeanDiff\tnumCpgs\tpMWU\tp2DKS\tmeanG1\tmeanG2" > $OUTDIR/MetileneDMR
metilene -a "H1-hESC" -b "leftVentricle" $SCRATCH/methylDackelOutputChr22/mergedOutputs_unionbedg.bed >>
```

Supplementary information

How can we explore hundreds of genomic regions for specific features, enrichments, etc.?

- DeepTools has a program called *computeMatrix*
- computeMatrix accepts signal files (e.g. in bigWig) and genomic region annotations (e.g. in BED or GTF) to calculate summary statistics
- computeMatrix has two modules:
  - reference-point: Obtains measures for entries of BED file (as reference) as well as their upstream and downstream
  - scale-regions: Calculates summary measures for BED files by shrinking each entry to a user-defined length

What is the methylation signal around CpG islands?

- Write a script to execute computeMatrix reference-point on CpG island BED file and the four bigWig files

```
MAINDIR=$SCRATCH/methylDackelOutputChr22
SAMPLES=($(ls $MAINDIR))
BWS=()
for SAMPLE in ${SAMPLES[@]}
do
  BW=$(ls $MAINDIR/$SAMPLE | grep bigWig)
  BWS+=($MAINDIR/$SAMPLE/$BW)
done
module load anaconda2/5.1.0 deeptools/3.2.1-anaconda2
OUTDIR=$SCRATCH/methylationMatricesChr22
mkdir -p $OUTDIR
computeMatrix reference-point -R $SCRATCH/Datasets/hg38CpgIslandsForChr22.bed -S ${BWS[@]} -o $OUTDIR/m
plotProfile -m $OUTDIR/mergedMethylationAroundIslands.tsv.gz -out $OUTDIR/mergedMethylationAroundIslands
plotProfile -m $OUTDIR/mergedMethylationAroundIslands.tsv.gz --perGroup --plotType heatmap -out $OUTDIR/
```

Troubleshooting guide

Throughout this tutorial, you may see that you do not see the same output as the instructor.

Feel free to ask for help.

Some common reasons include:

- You are not logged into the teaching cluster.
  - You can type `echo $HOSTNAME` to see if it returns `teach01.scinet.local` or not
- Your session got disconnected and the environmental variables that you defined in the earlier steps are not initialized.
  - You can check if a variable is initialized by typing `echo $VARIABLENAME`. In the case of arrays, to see their elements, you can type `echo ${ARRAYNAME[@]}`.
  - If these are not initialized, nothing will be printed and that means you need to go back to the first occurence of the variable or array and re-run the command.