DNA Methylation Sequencing Analysis

I-Hsuan Lin

Published with GitBook

Table of Contents

Hands-on Training in Methylation Sequencing Analysis

I-Hsuan Lin

National Yang-Ming University, Taipei, Taiwan

*The materials was prepared for a NRPB workshop **(Hands-On Training in Methylation Sequencing Analysis)** and presented on 19th Dec 2014 at National Yang-Ming University.*

DNA methylation refers to the addition of a methyl group (-CH3) to the carbon atom on cytosine or to a nitrogen atom on adenine. Typical DNA methylation occurs at a cytosine base located 5' to a guanosine (i.e. CpG dinucleotide). Approximately 70% of the CpGs in the mammalian genome are methylated. However, dense clusters of CpG sites (known as CpG islands or CGI) are often un-methylated and found near transcriptional start sites (TSSs). The establishment, maintenance and erasing of cytosine methylation at regulatory regions can result in the modulation of gene expression, thus is one of the key epigenetic regulatory processes during development and disease.

Whole genome bisulfite sequencing (WGBS) is a high-throughput method that allows researchers to determine the pattern of DNA methylation across the entire genome. In Part 1 of this workshop, we demonstrated how to use the MethPipe methylation analysis pipeline to perform methylation calling at each CpG site and to identify HMRs (short hypomethylated regions) and PMDs (large partially methylated domains). In this session, we will show you how to make use of open source tools and data from public databases in the analysis of BS-seq data.

License

Data Preparation

First, log into your account at **alps1.nchc.org.tw** with the user name and password provided. As show in **Table 1**, you will find four folders in your home directory.

Table 1. Dolders in the home directory

Folder name	Description
Data	Downloaded public data
Output	Processed files
Scripts	R scripts and output PDF files
Tools	Executables

Things you need to know before using the ALPS server:

1. Users must submit your jobs to **LSF** via the **bsub** command. The escape character (or backslash "\") was added before special character to make the command compatible.
2. The allowable space for each account is 1GB.

Things you need to know if you are performing this tutorial on your own computer/server:

1. Most of the commands are written for batch job submission to the **LSF** system. You will need to modify the commands if you are not using LSF.
2. You may create the folders as listed in **Table 1** in your home directory, or other locations and remember to change the file paths in commands.

Locate the MethPipe Files

Several files MethPipe-associated files were already provided in the ALPS1 server and they can be found in the "**/work3/NRPB1219**" folder.

```
ls -lah /work3/NRPB1219/hg18_*
```

```
# Console output

-rw-r--r-- 1 s00yao00 s00yao00 1.7M 2014-12-16 15:55 /work3/NRPB121
-rw-r--r-- 1 s00yao00 s00yao00 777M 2014-12-16 15:55 /work3/NRPB121
-rw-r--r-- 1 s00yao00 s00yao00 163K 2014-12-16 15:55 /work3/NRPB121
-rw-r--r-- 1 s00yao00 s00yao00 1.7M 2014-12-17 11:51 /work3/NRPB121
-rw-r--r-- 1 s00yao00 s00yao00 797M 2014-12-16 16:30 /work3/NRPB121
-rw-r--r-- 1 s00yao00 s00yao00 169K 2014-12-17 11:51 /work3/NRPB121
-rw-r--r-- 1 s00yao00 s00yao00 2.7M 2014-12-16 15:55 /work3/NRPB121
-rw-r--r-- 1 s00yao00 s00yao00 778M 2014-12-16 15:55 /work3/NRPB121
-rw-r--r-- 1 s00yao00 s00yao00  83K 2014-12-16 15:55 /work3/NRPB121
-rw-r--r-- 1 s00yao00 s00yao00 2.7M 2014-12-17 11:51 /work3/NRPB121
-rw-r--r-- 1 s00yao00 s00yao00 796M 2014-12-16 16:29 /work3/NRPB121
-rw-r--r-- 1 s00yao00 s00yao00  80K 2014-12-17 11:51 /work3/NRPB121
```

For reader who do not have an account on the ALPS server, you may download these files from locations listed below using `wget`.

```
wget http://120.126.44.231/ycl6/workshop/hg18_h1_bsseq_methpipe.hm
wget http://120.126.44.231/ycl6/workshop/hg18_h1_bsseq_methpipe.me
wget http://120.126.44.231/ycl6/workshop/hg18_h1_bsseq_methpipe.pm
wget http://120.126.44.231/ycl6/workshop/hg18_imr90_bsseq_methpipe
wget http://120.126.44.231/ycl6/workshop/hg18_imr90_bsseq_methpipe
wget http://120.126.44.231/ycl6/workshop/hg18_imr90_bsseq_methpipe
gunzip hg18_*.gz
```

And you may use `awk` commands to transform *.meth.corrected files into bedGraph format, and* .hmr and *.pmd files to BED format.

```
awk -F $'\t' 'BEGIN { OFS=FS } { print $1,$2,$2+1,$5 }' hg18_h1_bss
awk -F $'\t' 'BEGIN { OFS=FS } { print $1,$2,$2+1,$5 }' hg18_imr90_
awk -F $'\t' 'BEGIN { OFS=FS } { density = $5/($3-$2)*1000; print $
awk -F $'\t' 'BEGIN { OFS=FS } { density = $5/($3-$2)*1000; print $
awk -F $'\t' 'BEGIN { OFS=FS } { density = $5/($3-$2)*100000; print
awk -F $'\t' 'BEGIN { OFS=FS } { density = $5/($3-$2)*100000; print
```

Download Utilities

After locating the MethPipe files that we will need later in the analysis work flow, we first download the executables and place them into the “Tools” folder. We will use them for data file manipulation. We use the `chmod` command to make the programs executable.

```
cd ~/Tools
wget http://hgdownload.cse.ucsc.edu/admin/exe/linux.x86_64/bedGraph
wget http://hgdownload.cse.ucsc.edu/admin/exe/linux.x86_64/bedToBig
chmod 755 bedGraphToBigWig bedToBigBed
```

Now check if you can access `bedtools` utilities by trying one of its programs, such as `sortBed -h`

For reader who do not have an account on the ALPS server, please download the bedtools utilities (latest version is 2.22.0 at the time of writing) and compile it with `make` . The compiled executables will be located in the “bedtools-2.22.0/bin” folder. You may set up the PATH environment variable so that you can use any of the bedtools programs by calling its name without having to write the full file path.

```
cd ~/Tools
wget -O bedtools-2.22.0.tar.gz 'https://github.com/arq5x/bedtools2/
tar zxf bedtools-2.22.0.tar.gz

cd bedtools2
make
```

Download Annotations

For this workshop, we have also downloaded all relevant annotation files from public databases (UCSC Genome Browser and Gencode) and placed them in the “/work3/NRPB1219” folder.

```
# Console output

-rw-r--r-- 1 s00yao00 s00yao00 2.2K 2008-09-05 18:13 /work3/NRPB12
-rw-r--r-- 1 s00yao00 s00yao00 1.6M 2006-04-14 02:39 /work3/NRPB12
-rw-r--r-- 1 s00yao00 s00yao00 653M 2014-12-16 16:01 /work3/NRPB12
-rw-r--r-- 1 s00yao00 s00yao00 289M 2014-10-20 02:54 /work3/NRPB12
```

For reader who do not have an account on the ALPS server, please download and uncompressed these files.

```
wget ftp://ftp.sanger.ac.uk/pub/gencode/release_3c/gencode.v3c.annc
wget http://hgdownload.soe.ucsc.edu/goldenPath/hg18/database/chromI
wget http://hgdownload.soe.ucsc.edu/goldenPath/hg18/database/cpgIsl
wget http://hgdownload.soe.ucsc.edu/goldenPath/hg18/database/wgEncc
gunzip gencode.v3c.annotation.NCBI36.gtf.gz chromInfo.txt.gz cpgIsl
```

The Repeating Elements information provided the by UCSC Genome Browser was split into individual chromosomes, i.e. one data file per one chromosome. We will use a shell script to demonstrate batch download and joining of the data files into one single file for ease manipulation.

```
cd ~/Data
wget --no-check-certificate https://raw.githubusercontent.com/ycl6/
sh download_rmsk.sh
```

Use `ls` to check the file was in the "Data" folder.

```
ls -la ~/Data/hg18.rmskRM327.bed.gz
```

```
# Console output

-rw------- 1 s00yao00 s00yao00 52052411 2014-12-20 19:39 /home/s00y
```

Annotation File Preparation – Defining Genomic Regions

In this section, we will use the reference annotation files retrieved from public databases to generate several genomic positional templates in BED format. We will use these files in the calculation of DNA methylation at specific regions, such as promoter, intergenic, CpG islands, etc.

500 Chromosomal Bins

We will use the `makewindows` program of the bedtools utilities to subset the genome into 500 bins for each chromosome.

```
cd ~/

bsub -q 16G -o stdout -e stderr "bedtools makewindows -g /work3/NRF
```

CpG Islands and Transcriptional Factor Binding Sites (TFBS)

The CpG island and TFBS annotation files downloaded from the UCSC genome browser is not in BED format. Hence we use `awk` to process them. We then pipe the output in the `sortBed` program of the bedtools utilities to make sure the BED files were properly coordinate-sorted.

```
cd ~/

awk -F $'\t' 'BEGIN { OFS=FS } { print $1,$2,$3,"CpG|"$6,$8*10,"+"

awk -F $'\t' 'BEGIN { OFS=FS } { if($6 >= 500) print $2,$3,$4,$5,$6
```

The above command will change the content of the files from:

```
# cpgIslandExt.txt
chr1    18598   19673   CpG: 116        1075    116     787     21
chr1    124987  125426  CpG: 30 439     30      295     13.7    67
chr1    317653  318092  CpG: 29 439     29      295     13.2    67
chr1    427014  428027  CpG: 84 1013    84      734     16.6    72
chr1    439136  440407  CpG: 99 1271    99      777     15.6    61

# wgEncodeRegTfbsClustered.txt
585     chr1    51      504     ZBTB33  392     .       51      504
585     chr1    130     306     c-Jun   459     .       130     306
585     chr1    247     292     NFKB    1000    .       247     292
585     chr1    7125    8886    GR      99      .       7125    888
585     chr1    9914    10042   HEY1    44      .       9914    100
```

And into BED format:

```
# cpgIslandExt.bed
chr1    18598   19673   CpG|116 216     +
chr1    124987  125426  CpG|30  137     +
chr1    317653  318092  CpG|29  132     +
chr1    427014  428027  CpG|84  166     +
chr1    439136  440407  CpG|99  156     +

# wgEncodeRegTfbsClustered.bed
chr1    247     292     NFKB    1000    +
chr1    81184   81397   Rad21   1000    +
chr1    227464  227809  Rad21   602     +
chr1    227505  227800  CTCF    1000    +
chr1    530491  530823  c-Myc   1000    +
```

Promoters, Exonic, Intronic, UTR and intergenic Regions

The gene annotation file we used here was curated by the GENCODE project (http://www.gencodegenes.org/). The GTF format is a standardized way to present gene annotation information of a genome (including gene, transcript, UTR, exon,

start and stop codon).

```
head -n 15 /work3/NRPB1219/gencode.v3c.annotation.NCBI36.gtf
```

```
# Console output

##description: evidence-based annotation of the human genome (NCBI3
##provider: GENCODE
##contact: fsk@sanger.ac.uk
##format: gtf
##date: 2009-10-01
chr1    HAVANA  gene    1737    4275    .       +       .       ger
chr1    ENSEMBL transcript      1737    4275    .       +       .
chr1    ENSEMBL exon    1737    2090    .       +       .       ger
chr1    ENSEMBL exon    2476    2584    .       +       .       ger
chr1    ENSEMBL exon    3084    4275    .       +       .       ger
chr1    ENSEMBL CDS     4022    4249    .       +       0       ger
chr1    ENSEMBL start_codon     4022    4024    .       +       0
chr1    ENSEMBL stop_codon      4250    4252    .       +       0
chr1    ENSEMBL UTR     1737    2090    .       +       .       ger
chr1    ENSEMBL UTR     2476    2584    .       +       .       ger
```

Please refer to online documentation for more information about the GTF format:

- http://www.ensembl.org/info/website/upload/gff.html
- http://www.gencodegenes.org/gencodeformat.html

By executing the commands below, we will process the GTF file and generate genomic regions such as promoter regions, exonic, intronic, intergenic regions in BED format. The `sortBed` and `mergeBed` programs of the bedtools utilities was used to ensure the output is coordinate-sorted and overlapping regions were merged.

```
cd ~/

bsub -q 16G -o stdout -e stderr "awk -F $'\t' 'BEGIN { OFS=FS } { i
```

!!! Use `bjobs` to check the job has completed, and `ls` to make sure this file was created before proceed.

```
ls -la Data/gencode.v3c.exon_merged.bed.gz
```

```
# Console output

-rw------- 1 s00yao00 s00yao00 2097963 2014-12-19 23:24 Data/gencod
```

```
cd ~/

bsub -q 16G -o stdout -e stderr "awk -F $'\t' 'BEGIN { OFS=FS } { i

bsub -q16G -o stdout -e stderr "awk -F $'\t' 'BEGIN { OFS=FS } { if

bsub -q 16G -o stdout -e stderr "awk -F $'\t' 'BEGIN { OFS=FS } { i
```

!!! Use `bjobs` to check the all jobs have completed, and `ls` to make sure this file was created before proceed.

```
ls -la Data/gencode.v3c.transcript.bed.gz
```

```
# Console output

-rw------- 1 s00yao00 s00yao00 1478470 2014-12-19 23:33 Data/gencod
```

The promoter region defined here is the transcription start site (TSS) upstream 1000 bp & TSS downstream 500 bp.

```
cd ~/

bsub -q 16G -o stdout -e stderr "zcat Data/gencode.v3c.transcript.b
```

The `print_utr.pl` is adapted from http://davetang.org/muse/2013/01/18/defining-genomic-regions/.

```
cd ~/

bsub -q 16G -o stdout -e stderr "perl /work3/NRPB1219/print_utr.pl
```

Use `bjobs` to check the all jobs have completed and `ls` to see the required files were generated and placed in the "Data" folder.

`ls -la Data/gencode.*.gz`

```
# Console output

-rw------- 1 s00yao00 s00yao00 2097963 2014-12-17 10:14 Data/gencod
-rw------- 1 s00yao00 s00yao00  277528 2014-12-17 10:14 Data/gencod
-rw------- 1 s00yao00 s00yao00 1905992 2014-12-17 10:18 Data/gencod
-rw------- 1 s00yao00 s00yao00  535447 2014-12-17 10:25 Data/gencod
-rw------- 1 s00yao00 s00yao00 1478470 2014-12-17 10:23 Data/gencod
-rw------- 1 s00yao00 s00yao00 1792723 2014-12-17 10:29 Data/gencod
```

Analysis Work Flow

In this section, we will show readers how to use the `intersectbed` program of the bedtools utilities to identify the overlapped features in two sets of genomic features. Additionally, the `groupBy` program is then used to summarized statistics on selected data column(s) based upon common column groupings.

In DNA methylation analysis, this may be the mean methylation levels of promoter regions, whether the HMRs identified between two cells lines co-occur or are specific to each cell lines, or divide the HMRs into CpG island-containing and without CpG island.

DNA Methylation at Genomic Bins

For each one of the 500 bins, we use the `intersectBed` program of the bedtools utilities to compare the bins.bed file and meth.bedGraph files base-by-base. The output is piped into `groupBy` to calculate the number of features being joined (in this case is the methylation values at CpG) and the average methylation level for each bin (i.e. -c 7,7 -o count,mean). Lastly, `awk` was used to print the methylation levels in four decimal places.

```
# Console output

# intersectBed
chr1    0       494500  chr1    468     469     0
chr1    0       494500  chr1    470     471     0.666667
chr1    0       494500  chr1    483     484     0.5
chr1    0       494500  chr1    488     489     1
chr1    0       494500  chr1    492     493     0.857143

# groupBy
chr1    0       494500  3973    0.051545
chr1    494500  989000  11539   0.29057
chr1    989000  1483500 19948   0.28879
chr1    1483500 1978000 16014   0.4603
chr1    1978000 2472500 17089   0.47023

# awk
chr1    0       494500  3973    0.0515
chr1    494500  989000  11539   0.2906
chr1    989000  1483500 19948   0.2888
chr1    1483500 1978000 16014   0.4603
chr1    1978000 2472500 17089   0.4702
```

```
cd ~/

bsub -q 16G -o stdout -e stderr "intersectBed -a Data/hg18.500bins.

bsub -q 16G -o stdout -e stderr "intersectBed -a Data/hg18.500bins.
```

Use `bjobs` to check the all jobs have completed and `ls` to check the files was in the "Output" folder.

```
ls -la ~/Output/hg18.500bins.*.meth
```

```
# Console output

-rw------- 1 s00yao00 s00yao00 737091 2014-12-20 16:26 /home/s00yao
-rw------- 1 s00yao00 s00yao00 737091 2014-12-20 16:26 /home/s00yao
```

DNA Methylation at CpG Islands

Same as Chapter 2.1, we used `intersectBed` and `groupBy` commands to calculate the average methylation values at each CpG island.

```
# Console output

# intersectBed
chr1    18598   19673   CpG|116 216     +       chr1    18598   185
chr1    18598   19673   CpG|116 216     +       chr1    18612   186
chr1    18598   19673   CpG|116 216     +       chr1    18614   186
chr1    18598   19673   CpG|116 216     +       chr1    18627   186
chr1    18598   19673   CpG|116 216     +       chr1    18636   186

# groupBy
chr1    18598   19673   116     0
chr1    124987  125426  30      0
chr1    317653  318092  29      0
chr1    427014  428027  84      0
chr1    439136  440407  99      0
```

```
cd ~/

bsub -q 16G -o stdout -e stderr "intersectBed -a Data/cpgIslandExt.

bsub -q 16G -o stdout -e stderr "intersectBed -a Data/cpgIslandExt.
```

Use `bjobs` to check the all jobs have completed and `ls` to check the files was in the "Output" folder.

```
ls -la ~/Output/cpgIslandExt.*.meth
```

```
# Console output

-rw------- 1 s00yao00 s00yao00 960543 2014-12-20 17:45 /home/s00yac
-rw------- 1 s00yao00 s00yao00 960543 2014-12-20 17:44 /home/s00yac
```

DNA Methylation at TFBS

Same as Chapter 2.1, we used `intersectBed` and `groupBy` commands to calculate the average methylation values at each TF binding site.

```
# Console output

# intersectBed
chr1    81184   81397   Rad21   1000    +       chr1    81320   813
chr1    81184   81397   Rad21   1000    +       chr1    81334   813
chr1    227464  227809  Rad21   602     +       chr1    227650  227
chr1    227464  227809  Rad21   602     +       chr1    227656  227
chr1    227464  227809  Rad21   602     +       chr1    227664  227

# groupBy
chr1    81184   81397   Rad21   2       0
chr1    227464  227809  Rad21   5       0.1
chr1    227505  227800  CTCF    5       0.1
chr1    530491  530823  c-Myc   26      0.26923
chr1    535929  536126  Rad21   13      0.15385
```

```
cd ~/

bsub -q 16G -o stdout -e stderr "intersectBed -a Data/wgEncodeRegT

bsub -q 16G -o stdout -e stderr "intersectBed -a Data/wgEncodeRegT
```

Use `bjobs` to check the all jobs have completed and `ls` to check the files was in the "Output" folder.

```
ls -la ~/Output/wgEncodeRegTfbsClustered.*.meth
```

```
# Console output

-rw------- 1 s00yao00 s00yao00 15299532 2014-12-20 18:18 /home/s00y
-rw------- 1 s00yao00 s00yao00 15299532 2014-12-20 18:18 /home/s00y
```

DNA Methylation at Various Genic Structure Regions

Again, we use `intersectBed` and `groupBy` commands to calculate the average methylation values at each designated region.

Intergenic Regions

```
# Console output

# intersectBed
chr1    0       1736    chr1    468     469     0
chr1    0       1736    chr1    470     471     0.666667
chr1    0       1736    chr1    483     484     0.5
chr1    0       1736    chr1    488     489     1
chr1    0       1736    chr1    492     493     0.857143

# groupBy
chr1    0       1736    97      0.19548
chr1    20972   24416   27      0
chr1    25944   42911   148     0.10567
chr1    44799   52810   39      0.56999
chr1    53750   58917   21      0.22781
```

```
cd ~/

bsub -q 16G -o stdout -e stderr "intersectBed -a Data/gencode.v3c.i

bsub -q 16G -o stdout -e stderr "intersectBed -a Data/gencode.v3c.i
```

Merged Promoter Regions

```
# Console output

# intersectBed
chr1    736     2372    chr1    747     748     0
chr1    736     2372    chr1    749     750     0
chr1    736     2372    chr1    765     766     0
chr1    736     2372    chr1    770     771     0
chr1    736     2372    chr1    776     777     0

# groupBy
chr1    736     2372    71      0
chr1    18416   20728   136     0
chr1    25436   26944   22      0
chr1    41911   43411   8       0.1875
chr1    51810   53310   14      0.65505
```

```
cd ~/

bsub -q 16G -o stdout -e stderr "intersectBed -a Data/gencode.v3c.p

bsub -q 16G -o stdout -e stderr "intersectBed -a Data/gencode.v3c.p
```

Merged Exonic Regions

```
# Console output

# intersectBed
chr1    1736    2090    chr1    1741    1742    0
chr1    1736    2090    chr1    1808    1809    0
chr1    1736    2090    chr1    1822    1823    0
chr1    2475    2584    chr1    2534    2535    0
chr1    2475    2584    chr1    2537    2538    0

# groupBy
chr1    1736    2090    3       0
chr1    2475    2584    3       0
chr1    3083    4692    32      0.0625
chr1    4832    4901    3       0
chr1    5658    5810    5       0
```

```
cd ~/

bsub -q 16G -o stdout -e stderr "intersectBed -a Data/gencode.v3c.e

bsub -q 16G -o stdout -e stderr "intersectBed -a Data/gencode.v3c.e
```

Intronic Regions

```
# Console output

# intersectBed
chr1    2090    2475    chr1    2127    2128    0
chr1    2090    2475    chr1    2133    2134    0
chr1    2090    2475    chr1    2140    2141    0
chr1    2090    2475    chr1    2167    2168    0
chr1    2090    2475    chr1    2180    2181    0

# groupBy
chr1    2090    2475    12      0
chr1    2584    2837    7       0
chr1    2915    3083    2       0.33333
chr1    4692    4832    3       0.41667
chr1    4901    5658    19      0
```

```
cd ~/

bsub -q 16G -o stdout -e stderr "intersectBed -a Data/gencode.v3c.
bsub -q 16G -o stdout -e stderr "intersectBed -a Data/gencode.v3c.
```

UTR Regions

```
# Console output

# intersectBed
chr1    1737    2090    5_UTR   ENST00000456328 +       chr1    174
chr1    1737    2090    5_UTR   ENST00000456328 +       chr1    180
chr1    1737    2090    5_UTR   ENST00000456328 +       chr1    182
chr1    2476    2584    5_UTR   ENST00000456328 +       chr1    253
chr1    2476    2584    5_UTR   ENST00000456328 +       chr1    253

# groupBy
chr1    1737    2090    5_UTR   ENST00000456328 +       3       0
chr1    2476    2584    5_UTR   ENST00000456328 +       3       0
chr1    3084    4021    3_UTR   ENST00000456328 +       14      0.1
chr1    4226    4561    3_UTR   ENST00000438504 -       7       0
chr1    4226    4692    3_UTR   ENST00000423562 -       15      0
```

```
cd ~/

bsub -q 16G -o stdout -e stderr "intersectBed -a Data/gencode.v3c.U

bsub -q 16G -o stdout -e stderr "intersectBed -a Data/gencode.v3c.U
```

Check output files

Use `bjobs` to check the all jobs have completed and `ls` to check the files was in the "Output" folder.

```
ls -la ~/Output/gencode.v3c.*
```

```
# Console output

-rw------- 1 s00yao00 s00yao00 7640285 2014-12-20 18:23 /home/s00ya
-rw------- 1 s00yao00 s00yao00 7640285 2014-12-20 18:23 /home/s00ya
-rw------- 1 s00yao00 s00yao00 1048190 2014-12-20 18:25 /home/s00ya
-rw------- 1 s00yao00 s00yao00 1048190 2014-12-20 18:25 /home/s00ya
-rw------- 1 s00yao00 s00yao00 7621155 2014-12-20 18:25 /home/s00ya
-rw------- 1 s00yao00 s00yao00 7621155 2014-12-20 18:25 /home/s00ya
-rw------- 1 s00yao00 s00yao00 2100589 2014-12-20 18:23 /home/s00ya
-rw------- 1 s00yao00 s00yao00 2100589 2014-12-20 18:23 /home/s00ya
-rw------- 1 s00yao00 s00yao00 9696115 2014-12-20 18:36 /home/s00ya
-rw------- 1 s00yao00 s00yao00 9696115 2014-12-20 18:24 /home/s00ya
```

DNA Methylation at Repeat Elements

The BED file containing the repeat element annotations (prepared in Chapter 1.3) was located in the “Data” folder. In the hg18 version of the RepMask 3.2.7 annotation, the repeat elements were categorized into 21 repeat classes (see **Table 2**). In this demonstration, we will calculate an compare the methylation levels of five types common repeats: SINE, LINE, LTR, Satellite and DNA.

Table 2. Number of features in each repeat class in the RepMask 3.2.7 annotation (build hg18)

No. of Entries	Repeat Classes
1757823	*SINE*
1468898	*LINE*
699087	*LTR*
454482	*DNA*
407205	Simple_repeat
364098	Low_complexity
6998	Unknown
6096	*Satellite*
4251	snRNA
3589	Other
2204	RC
1871	DNA?
1751	tRNA
1715	rRNA
1437	srpRNA
1296	scRNA
715	RNA
417	SINE?
123	LTR?
93	Unknown?
52	LINE?

Like before, we use `intersectBed` and `groupBy` commands to calculate the average methylation values at each repeat element from the five repeat classes.

```
# Console output

# intersectBed
chr1    468     1310    Satellite       0       -       telo    TAR
chr1    468     1310    Satellite       0       -       telo    TAR
chr1    468     1310    Satellite       0       -       telo    TAR
chr1    468     1310    Satellite       0       -       telo    TAR
chr1    468     1310    Satellite       0       -       telo    TAR

# groupBy
chr1    468     1310    Satellite       89      0.21305
chr1    1540    1643    DNA     1       0
chr1    5128    5208    SINE    1       0
chr1    8769    8911    LINE    2       0
chr1    9877    10268   LINE    8       0
```

```
cd ~/

bsub -q 16G -o stdout -e stderr "intersectBed -a Data/hg18.rmskRM32

bsub -q 16G -o stdout -e stderr "intersectBed -a Data/hg18.rmskRM32
```

Use `bjobs` to check the all jobs have completed and `ls` to check the files was in the "Output" folder.

```
ls -la ~/Output/hg18.rmskRM327.*.meth
```

```
# Console output

-rw------- 1 s00yao00 s00yao00 119358409 2014-12-20 19:57 /home/s00
-rw------- 1 s00yao00 s00yao00 119358409 2014-12-20 19:57 /home/s00
```

Add CpG Islands Co-localization Information to HMR BED Files

Here, we will prepare the HMR BED file to include CpG island information using `intersectBed` and `groupBy` commands.

```
# Console output

# intersectBed
chr1    554333  558188  HYPO0   3855    .       0       0
chr1    558271  560164  HYPO1   1893    .       0       0
chr1    703534  704946  HYPO2   1412    CpG|60  563     563
chr1    714082  714363  HYPO3   281     .       0       0
chr1    751975  753029  HYPO4   1054    CpG|115 1029    750

# groupBy
# col5 = HMR length; col6 = No. of CGI; col7 = CGI length; col8 = N
chr1    554333  558188  HYPO0   3855    1       0       0
chr1    558271  560164  HYPO1   1893    1       0       0
chr1    703534  704946  HYPO2   1412    1       563     563
chr1    714082  714363  HYPO3   281     1       0       0
chr1    751975  753029  HYPO4   1054    1       1029    750
```

```
cd ~/

bsub -q 16G -o stdout -e stderr "intersectBed -a /work3/NRPB1219/hg

bsub -q 16G -o stdout -e stderr "intersectBed -a /work3/NRPB1219/hg
```

Use `bjobs` to check the all jobs have completed and `ls` to check the files was in the "Output" folder.

```
ls -la ~/Output/hg18_*_hmr.cgi.bed
```

```
# Console output

-rw------- 1 s00yao00 s00yao00 2044351 2014-12-20 18:54 /home/s00ya
-rw------- 1 s00yao00 s00yao00 3303689 2014-12-20 18:54 /home/s00ya
```

Similarity and Differences of HMRs and PMDs from H1 and IMR90

In this section, we wish to prepare files that record the amount of similarity and differences of HMRs (and also PMDs) between the two cell lines (i.e. H1 and IMR90). There may be several ways to do this depending on what information you wish to present to your audience. Below, we showed you one such method.

The concept is that we use `cat` command to join the HMRs or PMDs from H1 and IMR90, coordinate-sorted the data with `sortBed` and merge any overlapping regions with `mergeBed`. Common regions between H1 and IMR90 will be merged to become a wider region in this process. Then, we pass these merged HMRs into two consecutive `intersectBed` commands to identify the number and amount of overlaps between merged HMRs and HMRs in H1 (as well as that in IMR90).

HMRs

Calculate the amount overlapped between H1 and IMR90 HMRs.

```
cd ~/

bsub -q 16G -o stdout -e stderr "cat /work3/NRPB1219/hg18_h1_hmr.be
```

The first part of the full command line above produces the merged HMR BED output. We use `cat` to join the HMRs from H1 and IMR90, sort with `sortBed` and merge feature regions using `mergeBed` .

```
cat /work3/NRPB1219/hg18_h1_hmr.bed
/work3/NRPB1219/hg18_imr90_hmr.bed | sortBed | mergeBed -i - | head
```

```
# Console output

chr1    552721  558206
chr1    558242  562180
chr1    703351  704946
chr1    714082  714363
chr1    739503  744016
chr1    751866  753677
chr1    783310  784282
chr1    794854  798791
chr1    800933  803634
chr1    829069  831267
```

Then, we pipe the output from the first part of the command to two consecutive `intersectBed` to identify the overlapped HMRs between H1 and merged output, and also IMR90 and merged output.

```
# Console output

# intersectBed
# col1~3: merged HMR; col4~10: H1; col11~17: IMR90
chr1    552721  558206  chr1    554333  558188  HYPO0   21      +
chr1    558242  562180  chr1    558271  560164  HYPO1   24      +
chr1    703351  704946  chr1    703534  704946  HYPO2   23      +
chr1    714082  714363  chr1    714082  714363  HYPO3   39      +
chr1    739503  744016  .       -1      -1      .       -1      .
chr1    751866  753677  chr1    751975  753029  HYPO4   59      +
chr1    783310  784282  .       -1      -1      .       -1      .
chr1    794854  798791  chr1    794854  795385  HYPO5   69      +
chr1    800933  803634  chr1    801137  802542  HYPO6   23      +
chr1    829069  831267  chr1    829069  831267  HYPO7   76      +
```

Use `head` to print the first 10 lines of the file to standard output..

```
head ~/Output/h1_imr90_hmr_coverage.bed
```

```
# Console output

# col4 = H1 HMR; col5 = H1 HMR coverage; col6 = IMR90 HMR; col7 = I
chr1    552721  558206  3855    0.70    5485    1.00
chr1    558242  562180  1893    0.48    3938    1.00
chr1    703351  704946  1412    0.89    1275    0.80
chr1    714082  714363  281     1.00    0       0.00
chr1    739503  744016  0       0.00    4513    1.00
chr1    751866  753677  1054    0.58    1811    1.00
chr1    783310  784282  0       0.00    972     1.00
chr1    794854  798791  531     0.13    3937    1.00
chr1    800933  803634  1405    0.52    2701    1.00
chr1    829069  831267  2198    1.00    1279    0.58
```

PMDs

Calculate the amount overlapped between H1 and IMR90 PMDs.

```
cd ~/

bsub -q 16G -o stdout -e stderr "cat /work3/NRPB1219/hg18_h1_pmd.be
```

The first part of the full command line above produces the merged PMD BED output. We use `cat` to join the PMDs from H1 and IMR90, sort with `sortBed` and merge feature regions using `mergeBed` .

```
cat /work3/NRPB1219/hg18_h1_pmd.bed
/work3/NRPB1219/hg18_imr90_pmd.bed | sortBed | mergeBed -i - | head
```

```
# Console output

chr1    9468    29468
chr1    118468  124468
chr1    554378  560468
chr1    828757  839593
chr1    842967  869372
chr1    882748  892862
chr1    896296  906791
chr1    919108  931096
chr1    943504  955027
chr1    957031  976543
```

Then, we pipe the output from the first part of the command to two consecutive `intersectBed` to identify the overlapped PMDs between H1 and merged output, and also IMR90 and merged output.

```
# Console output

# intersectBed
# col1~3: merged PMD; col4~10: H1; col11~17: IMR90
chr5    102737  107812  chr5    102737  107812  PMD3128 98      +
chr5    369466  375940  chr5    369466  375940  PMD3129 123     +
chr5    524506  667718  chr5    524579  536321  PMD3130 102     +
chr5    524506  667718  chr5    540844  549323  PMD3131 106     +
chr5    524506  667718  chr5    574176  582161  PMD3132 100     +
chr5    722732  728932  chr5    722732  728932  PMD3133 96      +
chr5    743822  749562  chr5    743822  749562  PMD3134 104     +
chr5    836617  838832  chr5    836617  838832  PMD3135 180     +
chr5    1056949 1064091 chr5    1056949 1064091 PMD3136 112     +
chr5    1150146 1365624 .       -1      -1      .       -1      .
```

Use `head` to print the first 10 lines of the file to standard output..

```
grep chr5 ~/Output/h1_imr90_pmd_coverage.bed | head
```

```
# Console output

chr5    102737  107812  5075    1.00    0       0.00
chr5    369466  375940  6474    1.00    0       0.00
chr5    524506  667718  11742   0.08    143212  1.00
chr5    524506  667718  8479    0.06    143212  1.00
chr5    524506  667718  7985    0.06    143212  1.00
chr5    722732  728932  6200    1.00    0       0.00
chr5    743822  749562  5740    1.00    0       0.00
chr5    836617  838832  2215    1.00    0       0.00
chr5    1056949 1064091 7142    1.00    0       0.00
chr5    1150146 1365624 0       0.00    215478  1.00
```

Visualization Using R

In the this chapter, we will demonstrate how to use the methylation information produced in Chapter 2 to produce figures that aids in our exploration of the methylation patterns in these two cell lines.

For reader who do not have an account on the ALPS server, please check if you have access to the R environment for statistical analysis in your computer or server. If not, please download and compile the software.

```
wget http://cran.csie.ntu.edu.tw/src/base/R-3/R-3.1.2.tar.gz
tar zxfv R-3.1.2.tar.gz
cd R-3.1.2
./configure
make
```

R is also available on many operating system, including Windows. Please visit its website at http://www.r-project.org/.

Install R Libraries

The R available on ALPS is version 3.1.1 (latest version is 3.1.2 at the time of writing). Type `R` to initial the R environment.

```
R
```

```
R version 3.1.1 (2014-07-10) -- "Sock it to Me"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-unknown-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Within the R console, we use the `install.packages()` command to installed all required R libraries.

```
install.packages(c("data.table","ggplot2","gridExtra","reshape","sc
```

You may encounter these messages telling you certain directory you do not have WRITE permission. Please type **"y"** to both questions to create a personal library in your home directory.

```
Warning in install.packages(c("data.table", "ggplot2", "gridExtra",
  'lib = "/pkg/biology/R/R-3.1.1/lib64/R/library"' is not writable
Would you like to use a personal library instead?  (y/n) y
```

```
Would you like to create a personal library
~/R/x86_64-unknown-linux-gnu-library/3.1
to install packages into?  (y/n) y
```

Use the `library()` command to lists all available packages in the libraries.

```
library()
```

```
Packages in library ‘/home/s00yao00/R/x86_64-unknown-linux-gnu-libr

chron                   Chronological objects which can handle date
                        and times
colorspace              Color Space Manipulation
data.table              Extension of data.frame
dichromat               Color Schemes for Dichromats
digest                  Create Cryptographic Hash Digests of R Obje
ggplot2                 An implementation of the Grammar of Graphic
gridExtra               functions in Grid graphics
gtable                  Arrange grobs in tables.
labeling                Axis Labeling
munsell                 Munsell colour system
proto                   Prototype object-based programming
reshape                 Flexibly reshape data.
reshape2                Flexibly Reshape Data: A Reboot of the Resh
                        Package.
scales                  Scale functions for graphics.
stringr                 Make it easier to work with strings.
```

Press `q` to exit the package list, and type `q()` command to exit the R console. Choose **"n"** to not save the workspace as an image.

```
q()
Save workspace image? [y/n/c]: n
```

Execute the R Scripts

Plot Overview

```
cd ~/Scripts
wget --no-check-certificate https://raw.githubusercontent.com/ycl6/

Rscript --vanilla plot_overview.R
```

```
ls -la ~/Scripts/Figure1.pdf
```

```
# Console output

-rw------- 1 s00yao00 s00yao00 119244 2014-12-20 21:04 /home/s00yao
```

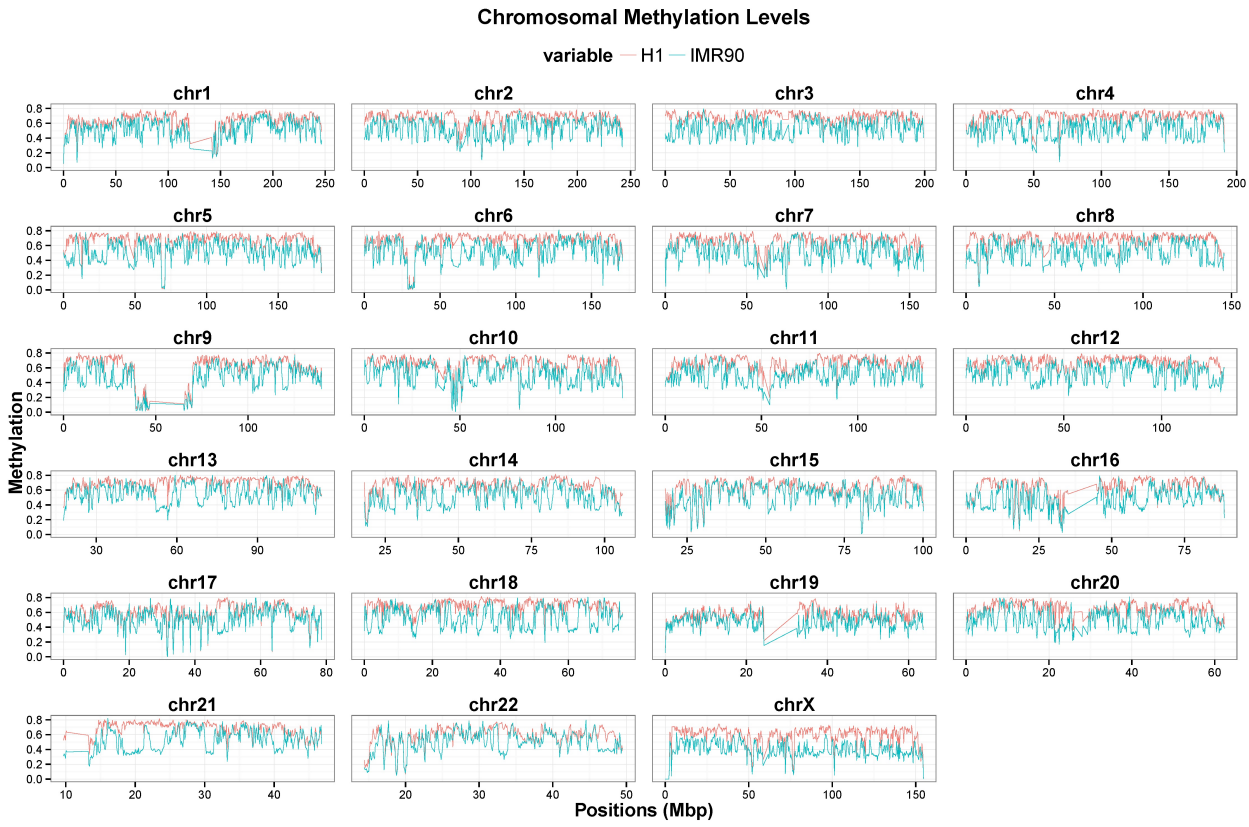


Figure 1. Line plots showing the DNA methylation in H1 and IMR90 along the chromosomal length.

Plot Locations

```
cd ~/Scripts
wget --no-check-certificate https://raw.githubusercontent.com/ycl6/

Rscript --vanilla plot_locations.R
```

```
ls -la ~/Scripts/Figure2*.pdf
```

```
# Console output

-rw------- 1 s00yao00 s00yao00     6025 2014-12-20 21:05 /home/s00y
-rw------- 1 s00yao00 s00yao00 14745034 2014-12-20 21:05 /home/s00y
-rw------- 1 s00yao00 s00yao00  1522371 2014-12-20 21:05 /home/s00y
```

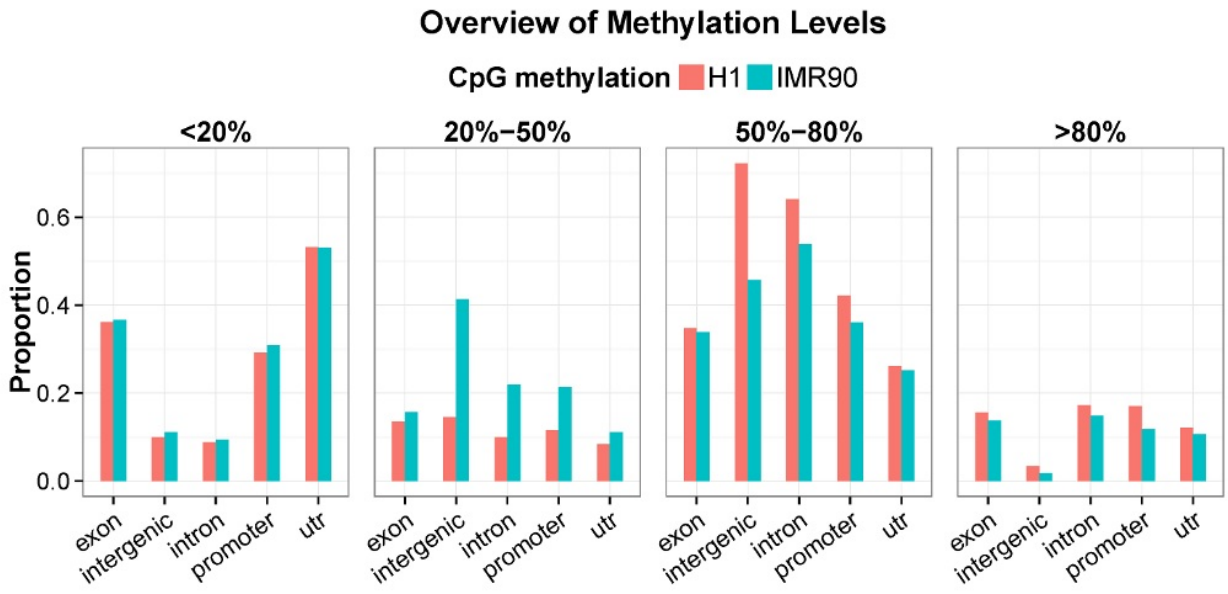


Figure 2A. Bar plots showing the DNA methylation levels within the five types of genomic regions. Majority of the differences between H1 and IMR90 occur in introns and intergenic regions that have between 20% to 80% DNA methylation levels.

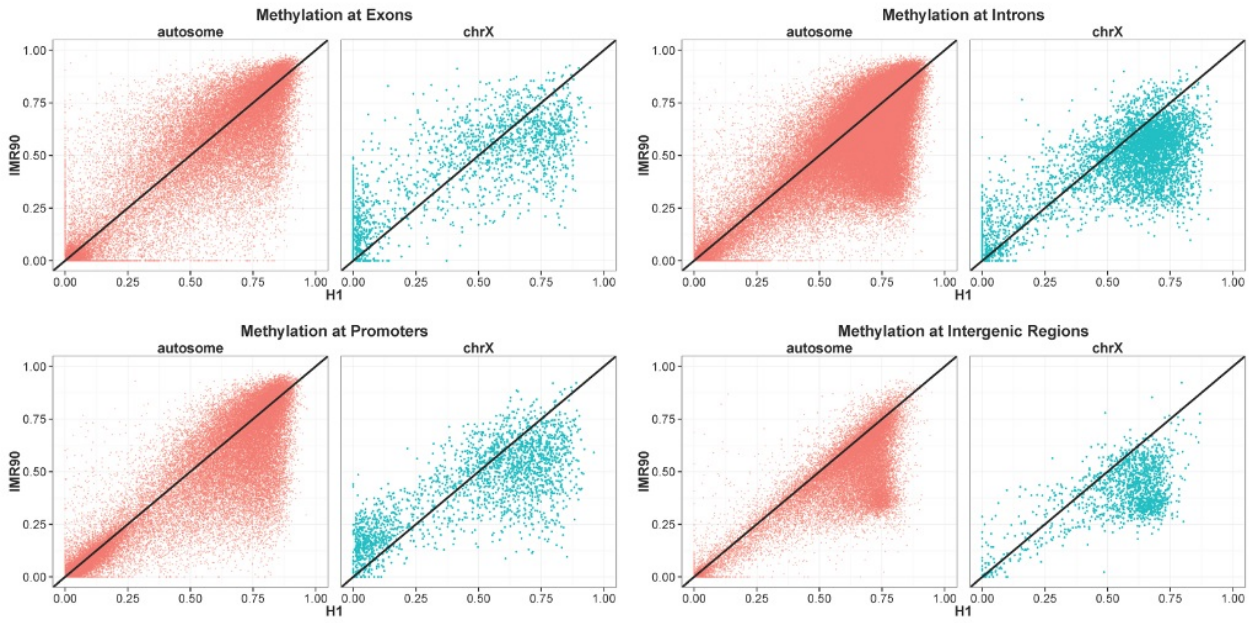


Figure 2B. H1 is more methylated than IMR90 in many regions (arrows). The DNA methylation pattern of X chromosome is also different between H1 (male) and IMR90 (female), especially in promoter regions (circle). This is due to the fact that female possesses two X chromosomes and one is randomly and permanently inactivated by DNA methylation.

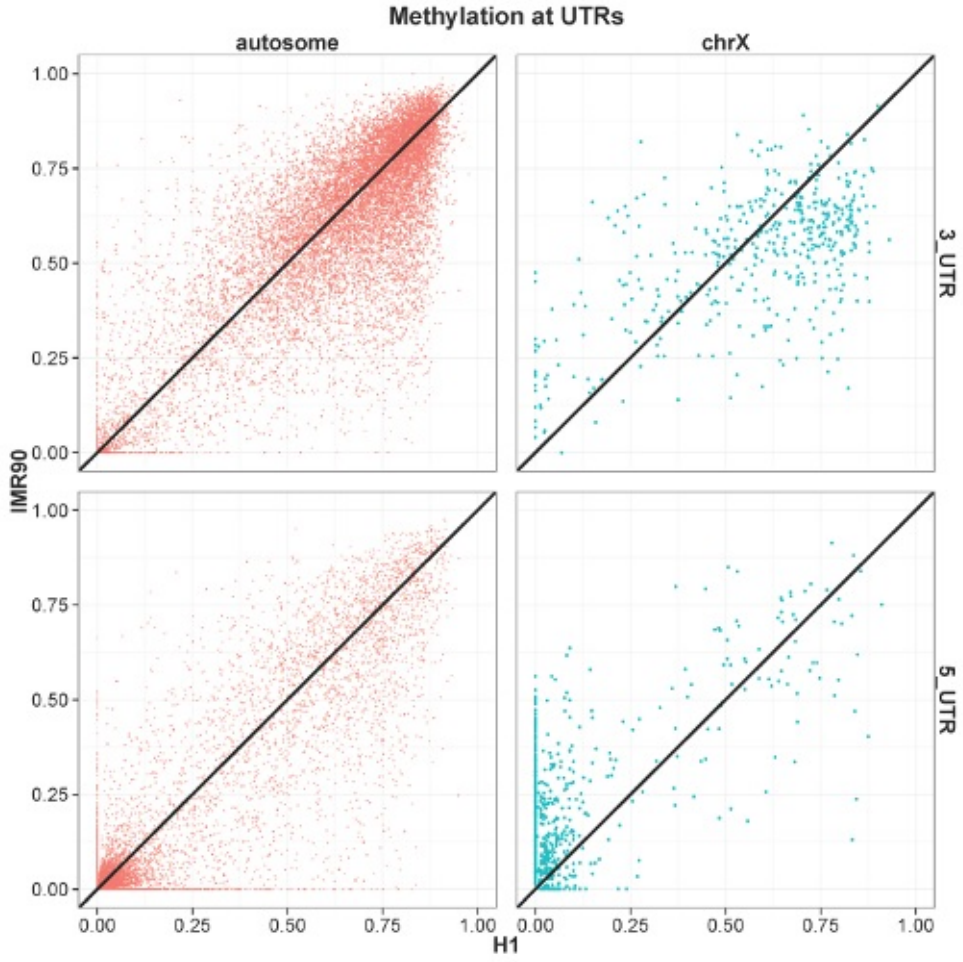


Figure 2C. The 5' UTRs are more hypomethylated compared to the 3' UTRs.

Plot Regulatory Sites

```
cd ~/Scripts
wget --no-check-certificate https://raw.githubusercontent.com/ycl6/

Rscript --vanilla plot_regulatory.R
```

```
ls -la ~/Scripts/Figure3.pdf
```

```
# Console output

-rw------- 1 s00yao00 s00yao00 13663 2014-12-20 21:06 /home/s00yao0
```

Table 3. Frequency table showing the CpG Island methylation levels.

Frequency Table		IMR90 (%)				
		<20%	20%-50%	50%-80%	>80%	
H1 (%)	<20%	67.6998	2.8443	0.3241	0.0000	70.87
	20%-50%	3.6164	7.3603	3.9114	0.0692	14.96
	50%-80%	0.8631	2.8006	8.1943	0.9943	12.85
	>80%	0.0291	0.0728	0.6118	0.6082	1.32
		72.21	13.08	13.04	1.67	
		IMR90 Total				H1 Total

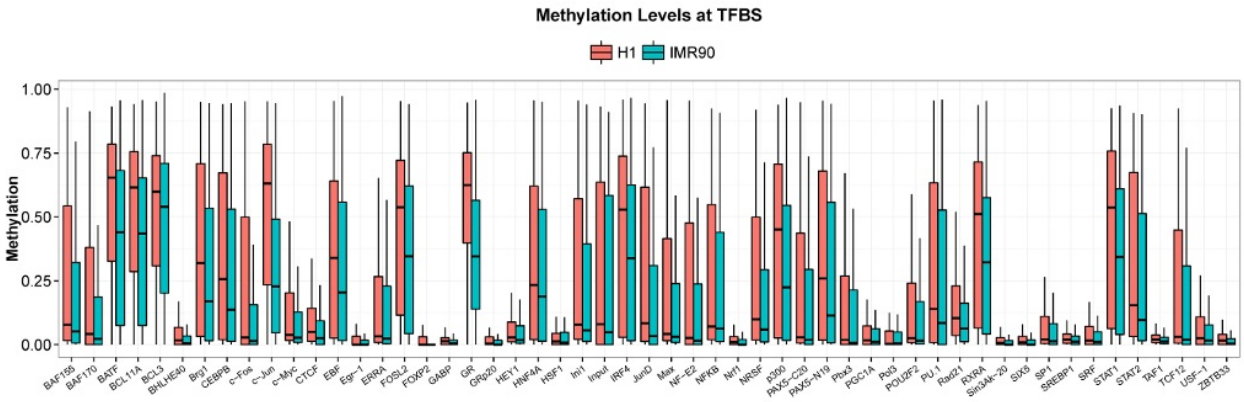


Figure 3A. Majority of the DNA methylation pattern between the two cell lines across all TFBS are similar, with a few exceptions such as c−Jun, GR, and p300. You may use t-test to test the significance of the difference between the two sample means.

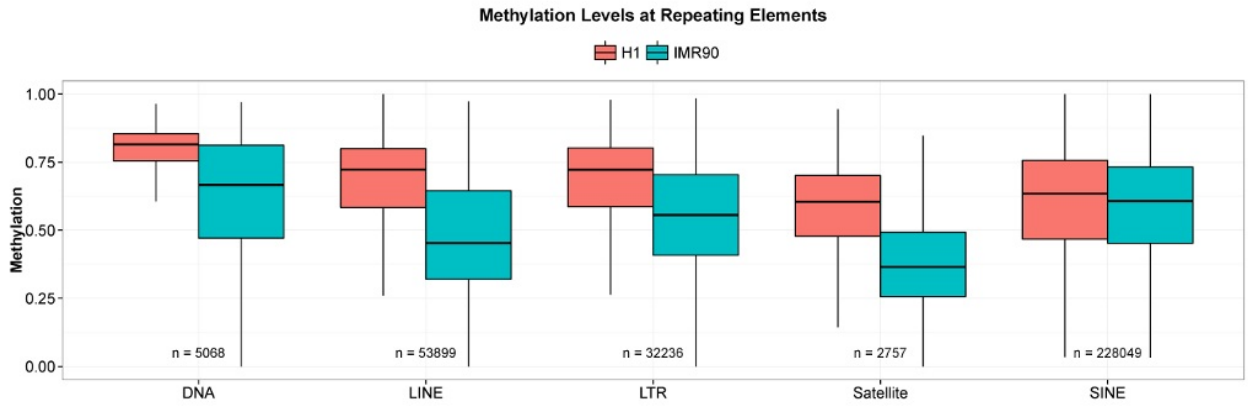


Figure 3B. Most of the repeated sequences in IMR90 are more hypomethylated except SINE elements.

Plot HMR and PMD Coverage

```
cd ~/Scripts
wget --no-check-certificate https://raw.githubusercontent.com/ycl6/

Rscript --vanilla plot_hmr_pmd.R
```

```
ls -la ~/Scripts/Figure4.pdf
```

```
# Console output

-rw------- 1 s00yao00 s00yao00 53267 2014-12-20 21:06 /home/s00yao0
```

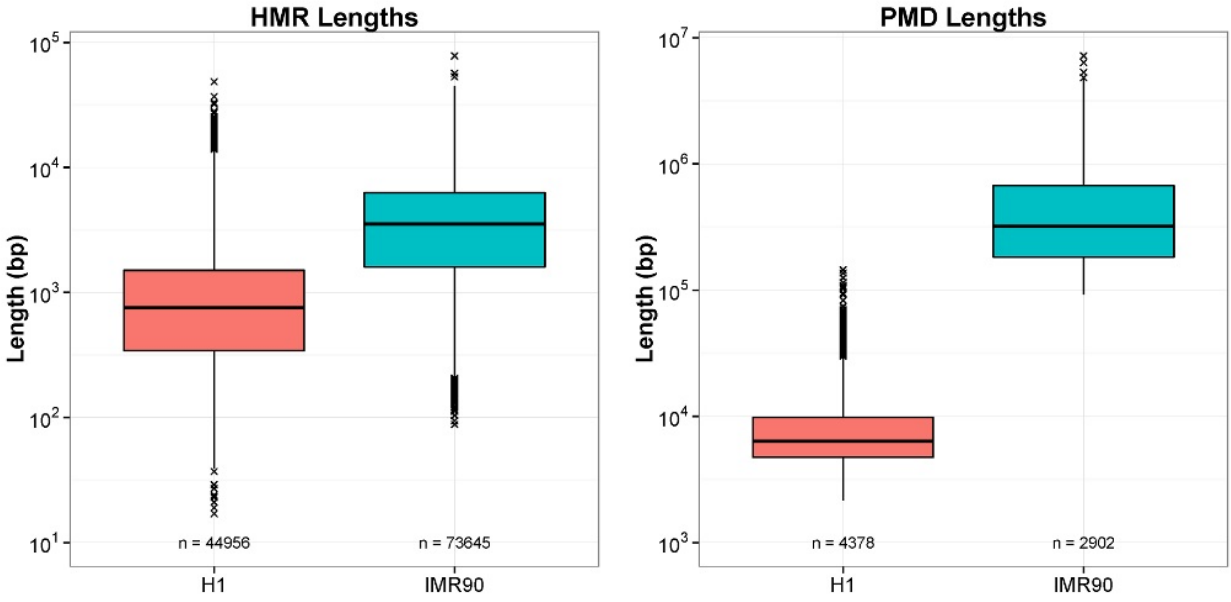


Figure 4A. Both the HMRs and PMDs are longer in IMR90. The PMD of IMR90 are approximately 50 times longer than that in H1 (PMD median length: ~ 320 Kbp in IMR90 and ~6Kbp H1)

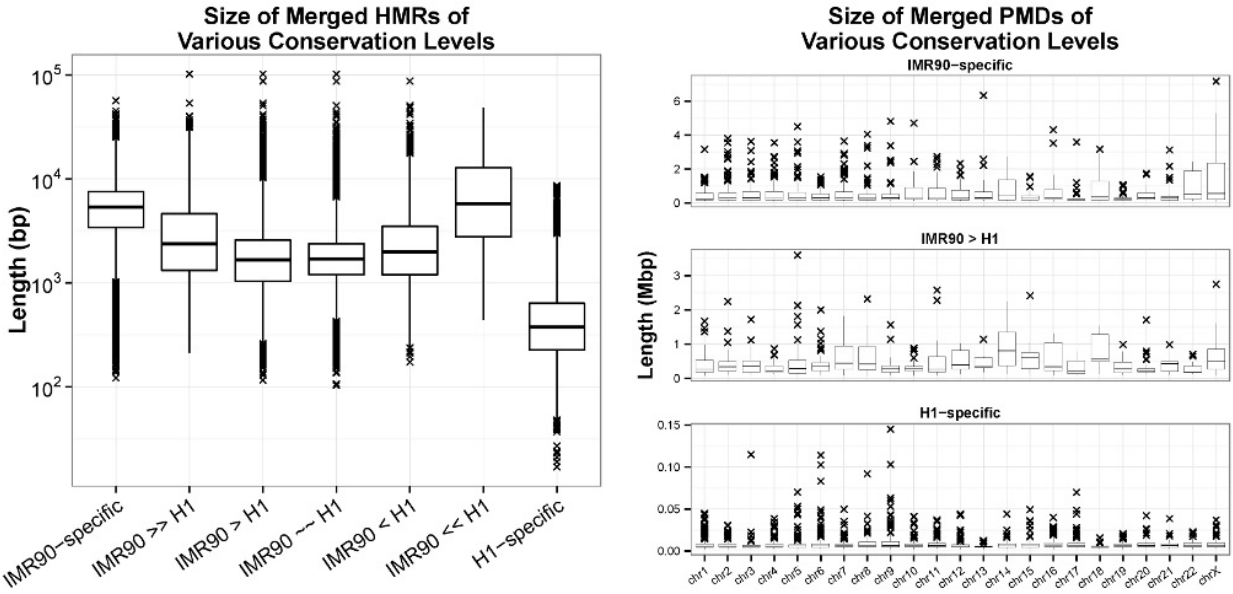


Figure 4B. The H1-specific HMRs are much shorter than other kinds of HMRs. The H1-specific PMDs are also very short compared to PMDs specific to IMR90. This data is consistent with that observed in **Figure 4A**.

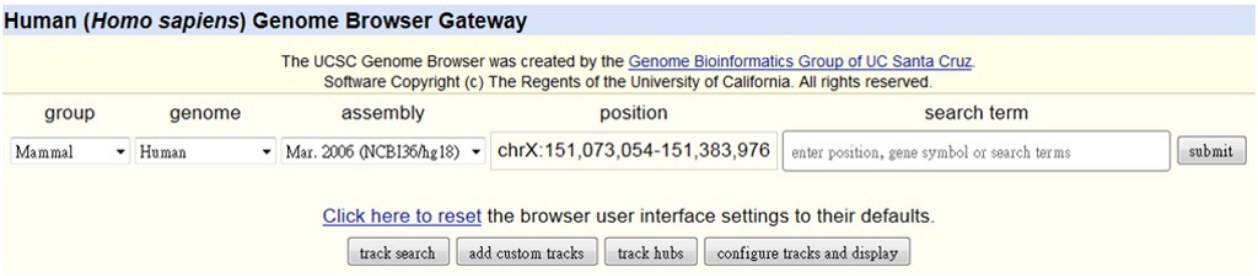
An Introduction of UCSC Genome Browser

> The UCSC Genome Browser (http://genome.ucsc.edu/) is developed and maintained by the Genome Bioinformatics Group, a cross-departmental team within the UC Santa Cruz Genomics Institute and the Center for Biomolecular Science and Engineering (CBSE) at the University of California Santa Cruz (UCSC).

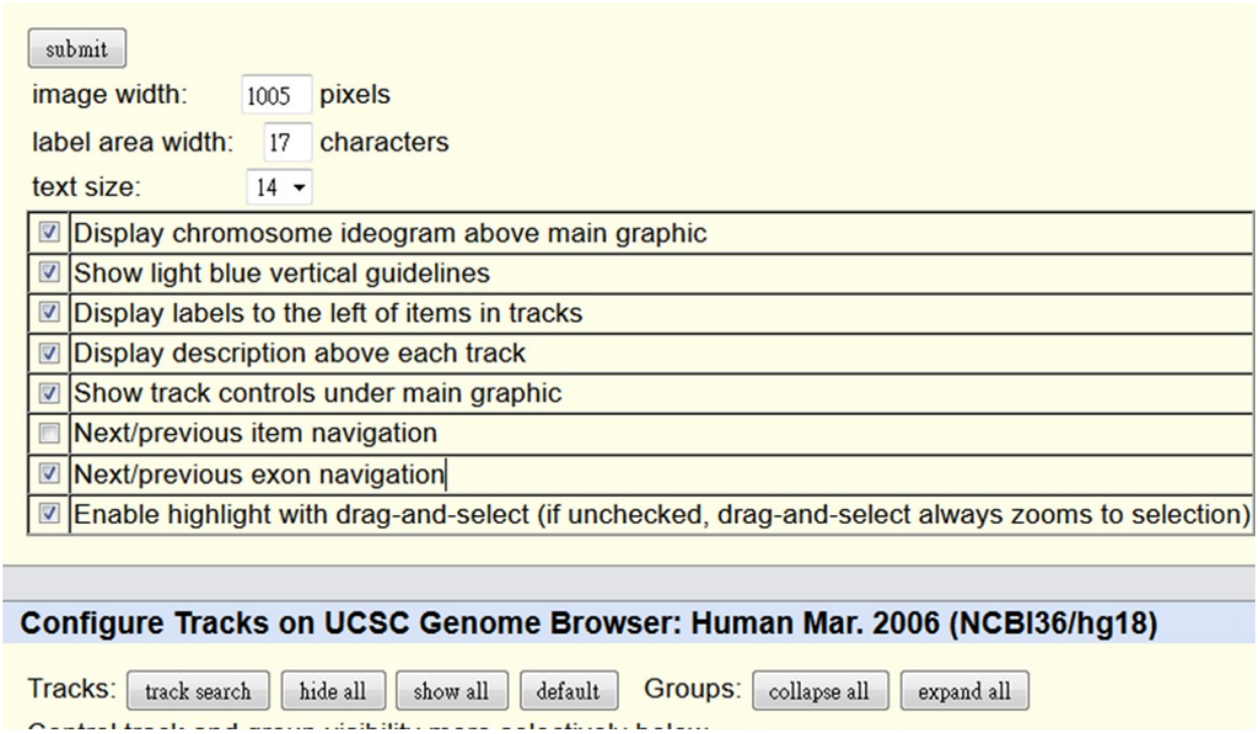
The UCSC Genome Browser provides a graphical interface to examine various data types (such as sequences, features, annotations, comparative analysis, etc.) from many genomes, ranging from yeast to humans. The straight-forward access to the wealth of data available via the Genome Browser is fundamental to genomic research.

General Usage

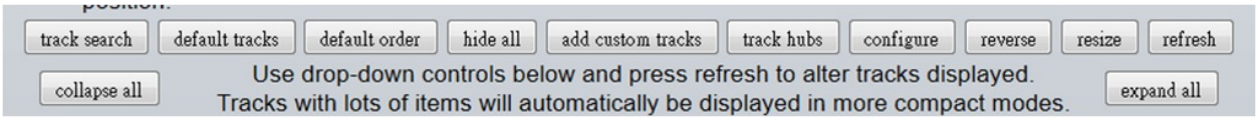
Access the UCSC Genome Browser via http://genome.ucsc.edu/cgi-bin/hgGateway. We will first choose the "Configure tracks and display" button to make some changes.



The default text size is **"8"** which is very small, we will change the size to **"12"**. You can also change the image to to fit your screen size, such as **"800"** pixel for smaller monitors, or **"1200"** or more for larger monitors. Click "Submit" to make the change.



When we go to the Browser page, click "Hide all" to close all data tracks. We will load the publically available DNA methylation data tracks using the **Track hubs** function.



Look for the **DNA Methylation** hub and click on "Connect" to add the this hub into our UCSC Genome Browser window.

Display	Hub Name	Description	Assemblies
Connect	Roadmap Epigenomics Data Complete Collection at Wash U VizHub	Roadmap Epigenomics Human Epigenome Atlas Data Complete Collection, VizHub at Washington University in St. Louis	hg19
Connect	UMassMed ZHub	UMassMed H3K4me3 ChIP-seq data for Autistic brains	hg19
Connect	Cancer genome polyA site & usage	An in-depth map of polyadenylation sites in cancer (matched-pair tissues and cell lines)	hg19
Connect	ENCODE Analysis Hub	ENCODE Integrative Analysis Data Hub	hg19
Connect	miRcode microRNA sites	Predicted microRNA target sites in GENCODE transcripts	hg19
Connect	Translation Initiation Sites (TIS)	Translation Initiation Sites (TIS) track	hg19
Connect	SDSU NAT	Sense/antisense gene/exon expression using Affymetrix exon array from South Dakota State University, USA	hg19, mm9, rn4
Connect	DNA Methylation	Hundreds of analyzed methylomes from bisulfite sequencing data	[+] hg19, hg18, mm9, mm10, panTro2, danRer7...

When we are directed back to the home page, choose **hg18** assembly, and key in **XIST** into the search term field and click "Submit".

Sometime the search terms may yield results from different gene annotation sources. We will choose the **XIST** gene from the **UCSC Genes** in this case. Feel free to choose other sources.

UCSC Genes

XIST (uc004ebm.1) at chrX:72957220-72989313 - Homo sapiens cDNA: FLJ21545 fis, clone COL06195.
TSIX (uc004ebn.2) at chrX:72928765-72965791 - Homo sapiens XIST antisense RNA (non-protein coding) (TSIX), non-coding RNA.

RefSeq Genes

XIST at chrX:72957211-72989313 - (NR_001564)

Non-Human RefSeq Genes

XIST at chrX:72957247-72988946 - (NR_001464)

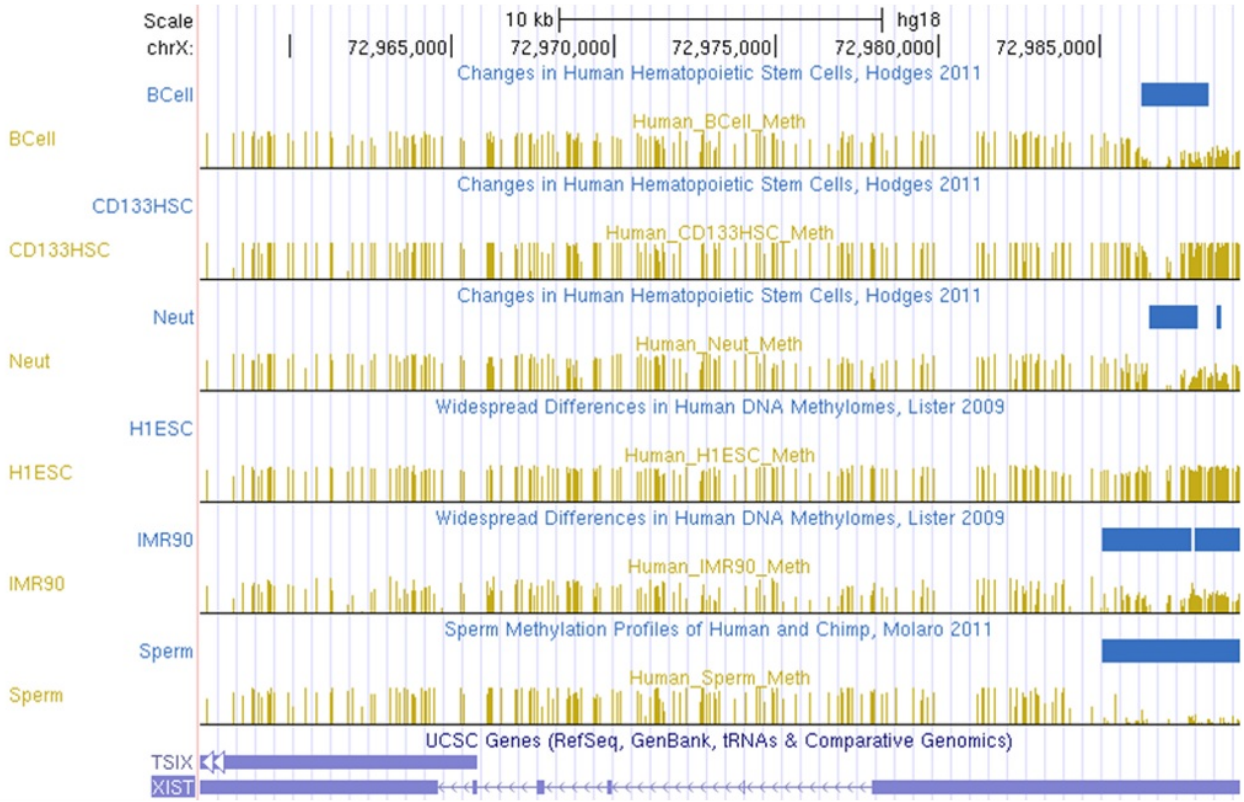
ENCODE Gencode Manual Gene Annotations (level 1+2) (Oct 2009)

XIST at chrX:72957211-72963015
XIST at chrX:72957213-72963366
XIST at chrX:72957216-72962861
XIST at chrX:72957216-72962904
XIST at chrX:72957216-72989313
XIST at chrX:72964269-72966027
XIST at chrX:72964361-72967770

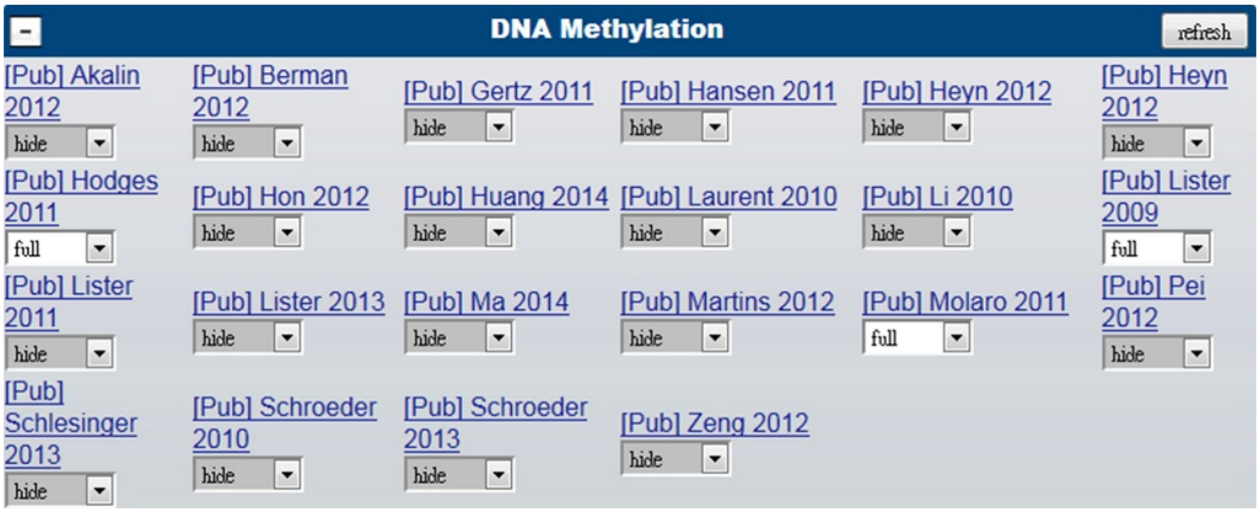
ENCODE Gencode Automated Gene Annotations (level 3) (Oct 2009)

XIST at chrX:72963932-72964336

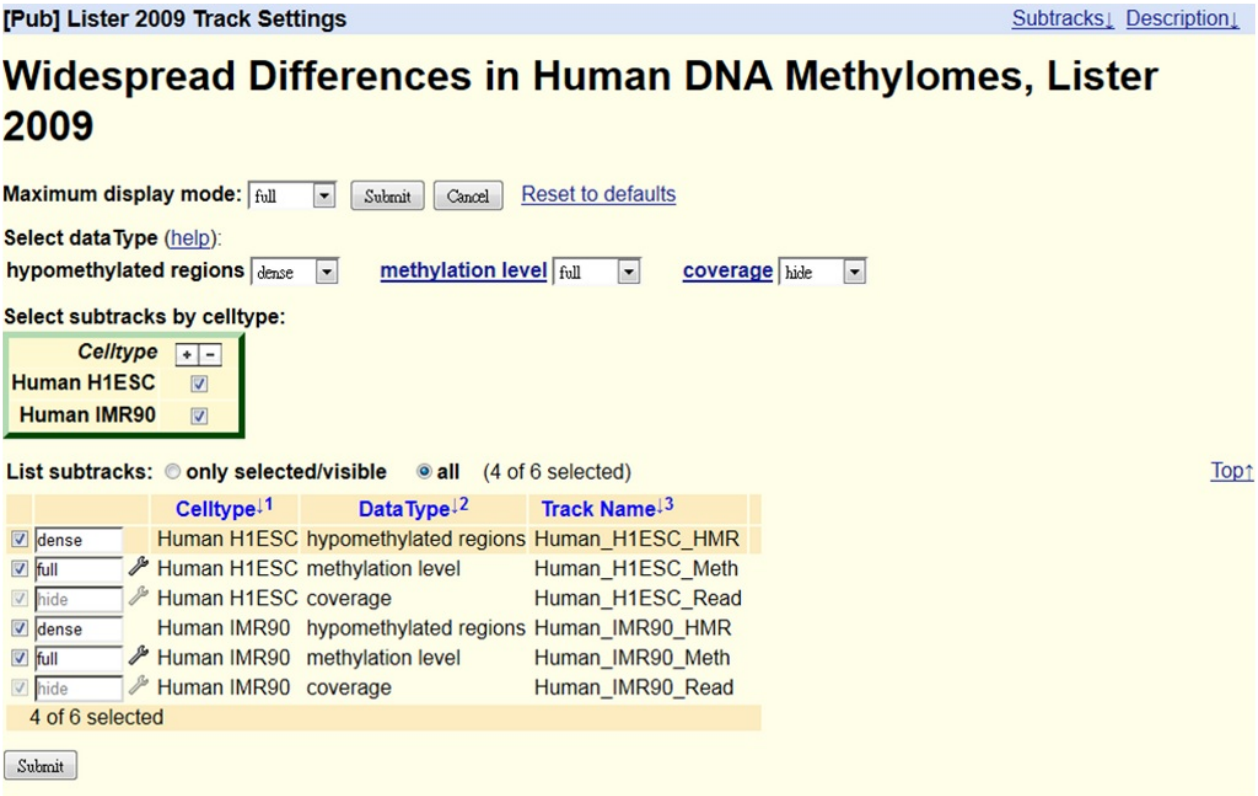
When directed back to the browser, we can now view the data tracks of several methylomes, and the **UCSC Genes** track below the methylation data tracks.



There are more DNA methylation tracks available in this public hub below the main view window.



By clicking on any one of the track title, such as **[Pub] Lister 2009**, you can change the mode of display and the types of tracks to show in the main window from this source. Feel free to try different settings.



The Compressed Binary Index Format

Due to the time limitation, we will not demonstrate how to set up your own track hub to host private data. Please refer to the UCSC Help page if you want to know more on this topic:

- http://genome.ucsc.edu/goldenpath/help/hgTrackHubHelp.html
- http://genome.ucsc.edu/goldenPath/help/customTrack.html

However, we will show here how to convert BED format to bigBed, and bedGraph to BigWIG. The file conversion process is necessary because UCSC Genome Browser only accepts compressed binary index formats that have smaller file sizes.

Using `bedGraphToBigWig`

```
cd ~/

bsub -q 16G -o stdout -e stderr "Tools/bedGraphToBigWig /work3/NRPE

bsub -q 16G -o stdout -e stderr "Tools/bedGraphToBigWig /work3/NRPE
```

Use `bjobs` to check the all jobs have completed and `ls` to check the files was in the "Output" folder.

`ls -la ~/Output/*.bw`

```
# Console output

-rw------- 1 s00yao00 s00yao00 269725108 2014-12-20 22:51 /home/s0(
-rw------- 1 s00yao00 s00yao00 272755992 2014-12-20 22:52 /home/s0(
```

Using `bedToBigBed`

```
cd ~/

bsub -q 16G -o stdout -e stderr "Tools/bedToBigBed /work3/NRPB1219/

bsub -q 16G -o stdout -e stderr "Tools/bedToBigBed /work3/NRPB1219/
```

Use `bjobs` to check the all jobs have completed and `ls` to check the files was in the "Output" folder.

`ls -la ~/Output/*.bb`

```
# Console output

-rw------- 1 s00yao00 s00yao00  936059 2014-12-20 22:50 /home/s00ya
-rw------- 1 s00yao00 s00yao00 1330574 2014-12-20 22:50 /home/s00ya
```