Assignment 4: Epigenomics

Start Early!

New Assignment Path Directory!

/storage1/fs1/workshops/Active/BIO5488/SP2024.L41.BIOL.5488.01/Assignments/week4/Assignment_Data/

Assignment Overview

- Explore methylation of CpGs
- Compare methylation patterns in promoter/non-promoter CpG islands

Reminders for Scripts

- Scripts should always start with shebang
- Must include docstring that:
  - Explains what the script does
  - Has a usage statement
- Import modules, e.g. sys and os
- Check for correct number of args

```
#!/usr/bin/env python3
```

```
"""
Takes input of raw RNA seq counts for before and after samples, filters, normalizes,
    calculates FLD for each gene, returns list of top ten genes with highest FLD
    score
Usage: python3 gene_expression.py raw_counts.txt
"""
```

```
import sys
```

```
# Check if correct number of args passed to script, if not exit and print
    documentation
if (len(sys.argv)!=2):
        sys.exit(__doc__)
```

BED Files

- Common file format for storing info on genomic features, annotations
- First three columns of a bed file are always: chr, start, end
- Remaining columns can contain any other information, e.g. sequences, coverage, strand, feature names, etc.
- Tab-delimited
  - Take this into consideration when reading and writing bed files
- Assignment instructions contain an appendix explaining data in each bed file we provide

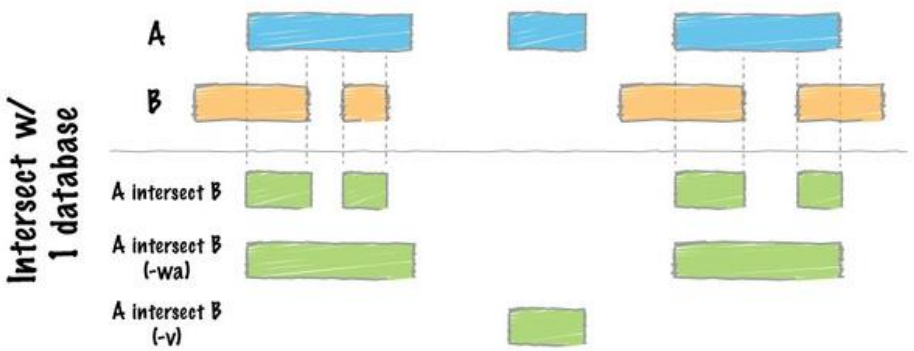
Example bed file

```
chr21 9411551 9411553
chr21  9411783  9411785
chr21 9412098 9412100
```

Check out the appendix for a description of each input file

bedtools

- Useful tool for manipulating bed files
  - https://bedtools.readthedocs.io/en/latest/
  - For assignment, should explore documentation for intersect, groupby, getfasta



Usage and option summary

Usage:

```
bedtools intersect [OPTIONS] -a <FILE> \
                             -b <FILE1, FILE2, ..., FILEN>
```

(or):

```
intersectBed [OPTIONS] -a <FILE> \
                       -b <FILE1, FILE2, ..., FILEN>
```

Option	Description
-a	BAM/BED/GFF/VCF file "A". Each feature in A is compared to B in search of overlaps. Use "stdin" if passing A with a UNIX pipe.
-b	One or more BAM/BED/GFF/VCF file(s) "B". Use "stdin" if passing B with a UNIX pipe. **NEW!!!**: -b may be followed with multiple databases and/or wildcard (*) character(s).
-abam	BAM file A. Each BAM alignment in A is compared to B in search of overlaps. Use "stdin" if passing A with a UNIX pipe: For example: samtools view -b <BAM> | bedtools intersect -abam stdin -b genes.bed. **Note**: no longer necessary after version 2.19.0
-ubam	Write uncompressed BAM output. The default is write compressed BAM output.
-bed	When using BAM input (-abam), write output as BED. The default is to write output in BAM when using -abam. For example: `bedtools intersect -abam reads.bam -b genes.bed -bed`
-wa	Write the original entry in A for each overlap.
-wb	Write the original entry in B for each overlap. Useful for knowing what A overlaps. Restricted by -f and -r.
-loj	Perform a "left outer join". That is, for each feature in A report each overlap with B. If no overlaps are found, report a NULL feature for B.
-wo	Write the original A and B entries plus the number of base pairs of overlap between the two features. Only A features with overlap are reported. Restricted by -f and -r.
-wao	Write the original A and B entries plus the number of base pairs of overlap between the two features. However, A features w/o overlap are also reported with a NULL B feature and overlap = 0. Restricted by -f and -r.
-u	Write original A entry once if any overlaps found in B. In other words, just report the fact at least one overlap was found in B. Restricted by -f and -r.

bedtools

- Useful tool for manipulating bed files
  - https://bedtools.readthedocs.io/en/latest/
  - For assignment, should explore documentation for intersect, groupby, getfasta

groupby

`bedtools groupby` is a useful tool that mimics the "group by" clause in database systems. Given a file or stream that is sorted by the appropriate "grouping columns" (`-g`), `groupby` will compute summary statistics on another column (`-c`) in the file or stream. This will work with output from all BEDTools as well as any other tab-delimited file or stream. As such, this is a generally useful tool for all command-line analyses, not just genomics related research.

Note

When using `bedtools groupby`, the input data must be ordered by the same columns as specified with the `-grp` argument, which establish which columns should be used to define a group of similar data. For example, if `-grp` is 1,2,3, the data should be pre-grouped accordingly. When `bedtools groupby` detects changes in the group columns it then summarizes all lines with that group. For example, `sort -k1,1 -k2,2 -k3,3 data.txt | bedtools groupby -g 1,2,3 -c 4 -o mean`.

Usage and option summary

Usage

```
bedtools groupby [OPTIONS] -i <input> -g <group columns> -c <op. column> -o <operation>
```

or:

```
groupBy [OPTIONS] -i <input> -g <group columns> -c <op. column> -o <operation>
```

Option	Description
-i	The input file that should be grouped and summarized. *Use "stdin" when using piped input.* **Note: if -i is omitted, input is assumed to come from standard input (stdin)**
-g (-grp)	Specifies which column(s) (1-based) should be used to group the input. Columns may be comma-separated with each column must be explicitly listed. Or, ranges (e.g. 1-4) are also allowed. *Default: 1,2,3*
-c (-opCol)	Specify the column (1-based) that should be summarized. *Required.*
-o (-op)	Specify the operation that should be applied to **opCol**. Valid operations: **sum** - *numeric only* **count** - *numeric or text* **count_distinct** - *numeric or text* **min** - *numeric only* **max** - *numeric only*

Overview

- Part 1.0: Examining Methylation from WGBS
- Part 1.1: Average CG Island Methylation
- Part 1.2: Plot Average CGI Methylation Dist.
- Part 1.3.0 (Step 1): Generating Promoters
- Part 1.3.0 (Step 2): Find Promoter, Non-Promoter CGIs
- Part 1.3.0 (Step 3): Analyze Average CpG Methylation in Promoter, Non-Promoter CGIs
- Part 1.3.0 (Step 4): Plot Average CGI Methylation Dist in Promoters, Non-Promoters
- Part 1.3.1: Calculate Frequency of CpGs in Promoter, Non-Promoter CGIs

- Extra Credit: (Examine H3K4me4 ChiP-Seq Data)
- Step 1: Calculate H3K4me4 RPKM in Promoters, Non-Promoters
- Step 2: Compare H3K4me3 RPKM Scores in Promoters, Non-Promoters

Part 1.0: Examining Methylation from WGBS

- BGM_WGBS.bed contains C and T coverage for each CpG
  - Reminder: WGBS converts unmethylated C's to T's
- Write a script **analyze_WGBS_methylation.py**
  - Calculate methylation level of each CpG, output bed file
  - Plot distribution for methylation levels
  - Plot coverage distribution for CpGs with 0X-100X coverage
  - Print fraction of CpGs with 0X coverage
- Make sure plots have axis labels, titles
- Do not hardcode output filenames

-

Part 1.1: Average CG Island Methylation

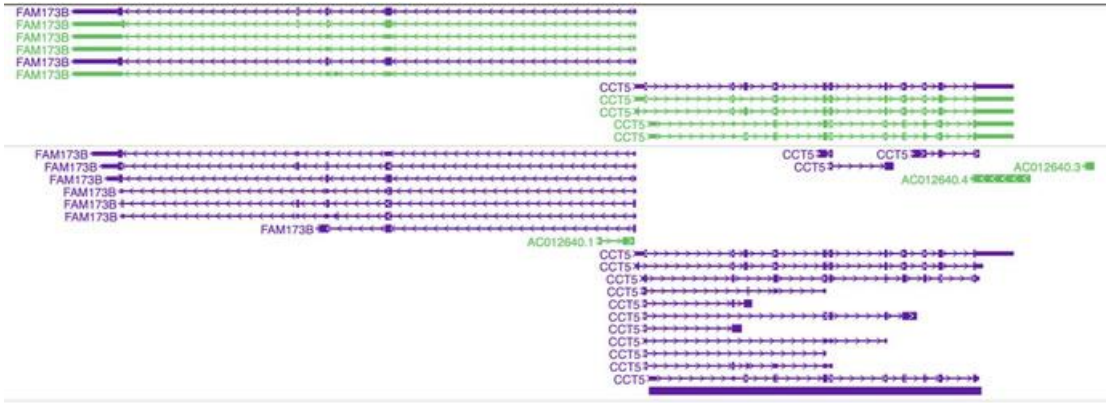
- Use CGI.bed, output bed file from previous step
- Calculate average CpG methylation in each CGI from CGI.bed
- Use bedtools for calculations
    - Look at intersect, groupby

Part 1.2: Plot Average CGI Methylation Dist.

- Use average CGI methylation bed created in previous step
- Write a script analyze_CGI_methylation.py
    - Plot distribution of average methylation levels
- Make sure plots have axis labels, titles
- Do not hardcode output filenames

Part 1.3.0 (Step 1): Generating Promoters

- Use refGene.bed
- Write a script **generate_promoters.py**
    - Generate bed file of promoter region coordinates
- Justify definition for choosing promoter coordinates (e.g. find literature source to support definition)
- Take strand (+/-) into consideration when determining promoter coordinates



Part 1.3.0 (Step 2): Find Promoter, Non-Promoter CGIs

- Use CGI.bed, bed file created in previous step
- Make two bed files
  - One for promoter CGIs
  - One for non-promoter CGIs
  - Use bedtools intersect
- Promoter CGIs mean CGIs that overlap promoter region

  Justify criteria for definition (# of bases) for overlapping

Part 1.3.0 (Step 3): Analyze Average CpG Methylation in Promoter, Non-Promoter CGIs

- Use promoter, non-promoter CGI bed files from previous step, WGBS CpG bed file generated in Part 1.0
- Calculate average CGI methylation for both bed files
- Use bedtools intersect, groupby
- Similar to commands for getting average methylation in Part 1.1

Part 1.3.0 (Step 4): Plot Average CGI Methylation Distin Promoters, Non-Promoters

- Use average CGI methylation files from previous step
- Run analyze_CGI_methylation.py(created in Part 1.2) on each file

Part 1.3.1: Calculate Frequency of CpGs in Promoter, Non-Promoter CGIs

- Use promoter, non-promoter CGI bed files
- Convert bed files to fasta files
  - Use bedtools getfasta

- Run **nuc_count_multisequence_fasta.py** on each fasta file
  - Provided in /storage1/fs1/workshops/Active/BIO5488/SP2024.L41.BIOL.5488.01/Assignments/week4/Assignment_Data/
  - Do NOT need to edit this script

What to Turn In

- Three scripts
  - analyze_WGBS_methylation.py
  - analyze_CGI_methylation.py
  - generate_promoters.py

- Seven bed files
  - BGM_WGBS_CpG_methylation.bed
  - WGBS_CGI_methylation.bed
  - refGene_promoters.bed
  - promoter_CGI.bed
  - non_promoter_CGI.bed
  - average_promoter_CGI_methylation.bed
  - average_non_promoter_CGI_methylation.bed

What to Turn In

- Five plots
  - BGM_WGBS_methylation_distribution.png
  - BGM_WGBS_CpG_coverage_distribution.png
  - WGBS_CGI_methylation_distribution.png
  - average_promoter_CGI_methylation.png
  - average_non_promoter_CGI_methylation.png

What to Turn In

- Completed README.txt file

Extra Credit: Examine H3K4me4 ChiP-Seq Data

Step 1: Calculate H3K4me4 RPKM in Promoters, Non-Promoters

- Use promoter, non-promoter CGI bed files (as feature files), BGM_H3K4me3.bed (provided in /storage1/fs1/workshops/Active/BIO5488/SP2024.L41.BIOL.5488.01/Assignments /week4/Assignment_Data/)
- Use **bed_reads_RKPM.pl** script
- Compare H3K4me3 signals in promoters vs non-promoters

Step 2: Compare H3K4me3 RPKM Scores in Promoters, Non-Promoters

- Write a script **analyze_H3K4me3_scores.py**
  - Plot two boxplots for H3K4me3 RPKM distribution in promoters, non-promoters on same figure

What to Turn In

- analyze_H3K4me3_scores.py
- H3K4me3_RPKM_promoter_CGI.bed
- H3K4me3_RPKM_non_promoter_CGI.bed
- H3K4me3_RPKM_promoter_CGI_and_H3K4me3_RPKM_non_promoter_CGI.png
- Answer additional questions in README.txt

Good luck!